

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Інститут прикладного системного аналізу
Кафедра математичних методів системного аналізу**

«На правах рукопису»
УДК 004.021

«До захисту допущено»
Завідувач кафедри
_____ О. Л. Тимошук
«__» _____ 20__ р.

**Магістерська дисертація
на здобуття ступеня магістра
зі спеціальності 124 Системний аналіз і управління
на тему: «Метод фотограмметрії дорожньої обстановки»**

Виконав:
студент II курсу, групи КА-61м
Краснощок Ілля Олександрович

Науковий керівник:
професор кафедри ММСА ІПСА, д.т.н., проф.
Данилов Валерій Якович

Рецензент:
професор кафедри інформаційної безпеки ФТІ
НТУУ «КПІ» імені Сікорського, д.т.н., проф.
Качинський Анатолій Броніславович

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних посилань.
Студент _____

Київ 2018

РЕФЕРАТ

Дисертацію виконано на 84 аркушах, вона містить 2 додатки та перелік посилань на використані джерела з 41 найменувань. У роботі наведено 42 рисунки та 20 таблиць.

Актуальність теми. На сьогоднішній день у світі спостерігається суттєве зростання інтересу до створення штучного інтелекту. Вже планується і вже частково впроваджено системи, які керують автомобілем без участі водія. Основним елементом даних систем є аналіз дорожньої обстановки, який, в основному, проводиться за допомогою пари стерео камер. Для вивчення положень об'єктів, їх параметрів використовується методи фотограмметрії, а саме стереофотограмметрія. За допомогою неї можна оцінювати відстань до об'єктів навколишнього середовища, дізнатися координати об'єкта в трьох вимірному просторі. Саме знаючи відстань до об'єктів, що оточують автомобіль, можна правильно прийняти рішення про подальший рух автомобіля, правильно здійснювати керування транспортним засобом, можливість прогнозувати подальший рух об'єктів та корегувати власну траєкторію руху.

Зв'язок роботи з науковими програмами, планами, темами. Дисертаційна робота виконувалась згідно з планом науково-дослідних робіт кафедри Математичних методів системного аналізу Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

Мета і задачі дослідження. Метою дисертаційної роботи є розробка методу оцінки положення об'єктів дорожньої обстановки за допомогою методів фотограмметрії.

Для досягнення вказаної мети було розв'язано такі задачі:

- провести аналіз методу фотограмметрії;
- провести аналіз основних проблем обробки фотографій для визначення невідповідності;

- розробити метод для оцінки глибини дорожньої обстановки;
- розробити систему для побудови карт глибини дорожньої обстановки зі стерео пар.

Об'єктом дослідження є дорожня обстановка.

Предметом дослідження є метод фотограмметрії.

Методи дослідження. Для розв'язання задачі використовувалися методи стереофотограмметрії (для розробки моделі побудови карти відстаней); методи оптимізації (для надання необхідної точності обраної моделі); методи теорії алгоритмів та програмування (для програмної реалізації розроблених алгоритмів); методи теорії ймовірності та математичної статистики (для проведення оцінки якості запропонованої моделі).

Наукова новизна одержаних результатів. Удосконалено модель дорожньої обстановки, яка, на відміну від існуючих, більш точно підраховує відстань до оточуючих об'єктів дорожньої обстановки.

Практичне значення одержаних результатів. Запропоновано метод, який може бути використано як помічник водію під час руху автомобіля або для руху автомобіля без участі водія. Розроблений метод, математичне та програмне забезпечення дозволяють швидко та якісно отримувати карту глибини дорожньої обстановки.

Публікації. Результати дисертації викладено в у міжнародному науковому журналі.

МЕТОД ФОТОГРАММЕТРІЇ, КАРТА ГЛИБИНИ, НЕВІДПОВІДНІСТЬ, ЗГОРТКОВА НЕЙРОННА МЕРЕЖА, КОРЕЛЯЦІЙНИЙ ШАР, ЕНКОДЕР-ДЕКОДЕР.

ABSTRACT

The thesis is presented in 84 pages. It contains 2 appendixes and bibliography of 41 references. 42 figures and 20 tables are given in the thesis.

Topic relevance. To date, in the world there is a significant increase in interest in the creation of artificial intelligence. Already planned and already partially introduced systems that control the car without the participation of the driver. The main element of these systems is the analysis of the road environment, which is mainly carried out using a pair of stereo cameras. To study the positions of objects, their parameters are used methods of photogrammetry, namely stereophotogrammetry. With it, you can estimate the distance to the objects of the environment, to find out the coordinates of the object in three-dimensional space. Just knowing the distance to the objects surrounding the car, you can correctly decide on the further movement of the car, correctly manage the vehicle, the ability to predict the further movement of objects and adjust their own trajectory of motion.

Thesis connection to scientific programs, plans, and topics. The thesis was prepared according to the scientific research plan of the Mathematical Methods of System Analysis Department of the National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute.”

Research goal and objectives. The goal of this thesis is to develop a method for evaluation the position of objects of the road with photogrammetry methods.

To accomplish this goal, the following objectives were reached:

- analyze photogrammetry method;
- analyze main problems of the processing of photographs for disparity calculation;
- develop a method for depth calculation of the road environment;
- develop a system for depth map generation of the road environment utilizing stereo pairs.

Object of research is road environment.

Subject of research is photogrammetric method.

Methods of research. To solve the task, the following methods were used: methods for stereophotogrammetry (for developing a model for mapping distances); methods of optimization (to provide the necessary accuracy of the chosen model); methods of algorithm theory and programming (for software implementation of developed algorithms); methods of probability theory and mathematical statistics (for evaluation of the quality of the proposed model).

Scientific contribution consists of the following: the model of the road environment, which, in contrast to the existing, more accurately calculates the distance to surrounding objects.

Practical value of obtained results. The proposed method can be used as a driver assistant or for self-driving cars. The developed method and software allow you to quickly and precisely receive a depth map of the road environment.

Publications. The results of the dissertation are presented in the international scientific journal.

PHOTOGRAMMETRY, DEPTH MAP, DISPARITY, CONVOLUTION NEURAL NETWORK, CORRELATION LAYER, ENCODER-DECODER

ЗМІСТ

| | |
|--|----|
| Перелік умовних позначень, скорочень і термінів | 6 |
| Вступ..... | 7 |
| 1 АНАЛІЗ ДОРОЖНЬОЇ ОБСТАНОВКИ | 11 |
| 1.1 Дорожня обстановка | 11 |
| 1.2 Проблеми аналізу зображень | 12 |
| Висновки до розділу | 20 |
| 2 МЕТОД ФОТОГРАММЕТРІЇ | 21 |
| 2.1 Основи фотограмметрії | 21 |
| 2.1.1 Предмет вивчення фотограмметрії | 21 |
| 2.1.2 Завдання фотограмметрії | 22 |
| 2.1.3 Класифікація фотограмметрії | 23 |
| 2.2 Систематика стерео відповідності | 24 |
| 2.2.1 Попередня обробка | 25 |
| 2.2.2 Функція вартості | 26 |
| 2.2.3 Сукупна функція вартості | 28 |
| 2.2.4 Покращення невідповідності | 39 |
| 2.3 Огляд літератури | 41 |
| 2.3.1 Навчання функції вартості | 41 |
| 2.3.2 Навчання кінець-в-кінець | 43 |
| Висновки до розділу | 43 |
| 3 ПОБУДОВА МОДЕЛІ ЗА ДОПОМОГОЮ МЕТОДУ ФОТОГРАММЕТРІЇ | 45 |
| 3.1 Згорткові нейронні мережі | 45 |
| 3.2 Кореляційний шар | 49 |
| 3.3 Підхід | 50 |
| 3.4 Тренування | 53 |
| 3.4.1 Датасети | 53 |

| | |
|--|-----|
| | 5 |
| 3.4.2 Метрика..... | 55 |
| Висновки до розділу | 56 |
| 4 РЕЗУЛЬТАТИ ПОБУДОВИ МЕТОДУ ОБЧИСЛЕННЯ КАРТИ ГЛИБИНИ | 57 |
| 4.1 Результати тренування | 57 |
| 4.2 Порівняння з іншими методами | 58 |
| Висновки до розділу | 61 |
| 5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ | 62 |
| 5.1 Опис ідеї проекту | 62 |
| 5.2 Технологічний аудит ідеї проекту..... | 65 |
| 5.3 Аналіз ринкових можливостей запуску стартап-проекту..... | 65 |
| 5.4 Розроблення ринкової стратегії проекту | 74 |
| 5.5 Розроблення маркетингової програми стартап-проекту..... | 77 |
| Висновки до розділу | 78 |
| ВИСНОВКИ..... | 79 |
| ПЕРЕЛІК ПОСИЛАНЬ | 80 |
| Додаток А Лістинг програми | 85 |
| Додаток Б Ілюстративний матеріал..... | 108 |

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

CNN — convolution neural network.

ЗНМ — згорткова нейронна мережа.

MAE — mean average error.

ВСТУП

Останнім часом, інтерес людства до самохідних автомобілів (не потребують участі людини в керуванні транспортним засобом) стрімко зріс.

Необхідною умовою аналізу дорожньої обстановки є положення автомобіля в просторі світу (дорожньої обстановки) та відносне положення транспортного засобу відносно інших об'єктів оточуючого середовища. Для цього в сучасних транспортних засобах для оцінки відстані до об'єктів зовнішнього середовища використовують системи на базі лазеру.

Дані системи є дуже дорогі, їх ціна може варіювати від 5 тис. до 60 тис. доларів США, що в результаті значно підвищить ціну на такий автомобіль.

Тому з 2016 року почали розглядати інші способи вимірювання відстані до інших об'єктів. Одним із основних способів вимірювання відстані до об'єктів базується на фотограмметрії – науці, що робить вимірювання з фотографій, з метою відновлення їх глибини. Фотограмметрія в своєму арсеналі має багато методів, але найбільш основним є стереофотограмметрія, що дозволяє відтворити глибину зображення лише з двох знімків – стерео пари.

З виходом відомої статті Alex Kryzhevskyi [1] 2012 року розвиток машинного навчання стрімко почав розвиватися. Технології машинного навчання почали використовувати в багатьох сферах людської діяльності.

З того часу розвиток комп'ютерного бачення набув нового значення і було розв'язано багато задач, які справлялися з подібною точністю як і людина [2].

Мета зіставлення двох стерео зображень – це оцінити значення невідповідності всіх пікселів лівого зображення відносно правого зображення стерео пари. Це одна з головних проблем комп'ютерного зору, яка має багато застосувань в прикладних задачах, такі як самохідні автомобілі, навігація роботів та в пристроях з доповненою реальністю. Хоча технічне визначення глибини, наприклад за допомогою LIDAR або ToF, легко дозволяє визначити глибину,

використання лише стерео пари для визначення глибини є більш ресурсозберігаючим підходом.

Маючи стерео пару, попередньо вирівнявши зображення, глибина пікселя може бути визначена зіставленням пікселів на стерео парі вздовж осі x при фіксованій осі y . Нехай для одного пікселя на лівому зображенні – (x, y) , відповідний піксель знаходиться з координатами $(x+d, y)$, d - невідповідність. Тоді глибина для даного пікселя може бути обчислена, як $\frac{f \cdot l}{d}$, де f – фокальна відстань камери, l – відстань між обома камерами. Так як глибина обернено пропорційна до невідповідності, то проблему знаходження глибини можна замінити на знаходження невідповідності.

Проблема стерео зіставлення інтенсивно досліджується з 1980-х років.

Проблеми визначення глибини: камери можуть по різному передавати колір пікселя (наприклад відсвічування сонця), зашумлення картинок, відмінність ракурсів у камер, регіони з однаковим кольором, повторення об'єктів (тин), дзеркальність об'єктів, прозорість (скло), загородження (на лівій камері об'єкт присутній, на іншій він затуляється іншим об'єктом).

Початкові дослідження в основному робили зіставлення в чотири кроки: розрахунок функції вартості, розрахунок сукупної функції вартості, розрахунок невідповідності та покращення невідповідності. До недавнього часу дані чотири кроки були присутні в багатьох методах для оцінки невідповідності [3,4,5,6]. Але останнім часом згорткові мережі показали свою продуктивність в багатьох задачах комп'ютерного зору [7,8,9]. Не винятком стала і задача покращення невідповідності в стерео парі. Використання згорткових мереж для оцінки глибини значно перевершило традиційні чотирьох крокові методи [10,11,12,31].

Вперше для стерео зіставлення згорткові нейронні мережі були застосовані Zbontar та Lecun [11] для підрахунку схожості двох пікселів на стерео парі. Даний метод досяг найкращої точності на Kitti2015 [17] і Middlebury [14] стерео датасетах в той час. Даний метод показав, що порівнювати значення пікселів у функції вартості було недостатньо, в той час коли згорткова нейронна мережа

може навчитися більш стійко знаходити невідповідність, використовуючи карти ознак зображення. Але в даній роботі були недоліки, а саме тривалий час підрахування невідповідності ($> 60s$); тому даний метод не можна використовувати для отримання невідповідності в закритих зонах (для точок, які не присутні на правому зображенні, але присутні на лівому); при використанні методів для покращення невідповідності використовуються параметри, що підбираються самостійно.

Дані роботи [15, 12] показали, що підрахунок функції вартості, підрахунок сукупної функції вартості та підрахунок невідповідності можуть бути порашовані згортковою нейронною мережею.

Зазвичай в традиційних методах комп'ютерного зору, наприклад, як Semi-Global Block Matching метод [16], використовується підрахунок функції енергії в декількох напрямках. На відміну від традиційних методів Mayer [15] і Kendal [12] використовують багато згорткових шарів, розташованих один за одним і тренували вихід даної мережі, щоб він був схожий на правильну невідповідність. Дані методи досягли найкращих результатів у точності та в часі отримання невідповідності, на відміну від методів, що використовували згорткові шари тільки для підрахунку функції вартості.

У статті Mayer та ін. [15], крім синтетичних датасетів, пропонується декілька варіантів згорткових мереж, які показали непогані результати на Kitti2015 [17]. Дані варіанти один від одного відрізняються, але найкращим виявився Dispnet з кореляційним шаром. Дана нейронна мережа відрізняється неймовірною швидкістю обрахунку невідповідності, близько 60 ms/frame. Також їхня архітектура побудована по типу – енкодер-декодер, що спочатку зменшує зображення, а потім збільшує його. Даний тип мереж зустрічається дуже часто, він присутній і у Kendal [12]. На відміну від мережі Mayer [15], ми додали ще шари в кінці їхньої архітектури, так щоб вихід мережі був таких самих розмірів як і вхід. У Mayer [15] вихід з мережі вдвічі менший, ніж вхід (на виході використовувалась інтерполяція до вхідного розміру зображення). Даний метод

досяг найкращої точності на Kitti2015 [17] і Middlebury [14] стерео датасетах в той час. Даний метод показав, що порівнювати значення пікселів у функції вартості було недостатньо, в той час коли згорткова нейронна мережа може навчитися більш стійко знаходити невідповідність, використовуючи карти ознак зображення. Але в даній роботі були недоліки, а саме тривалий час підрахування невідповідності ($> 60s$); тому даний метод не можна використовувати для отримання невідповідності в закритих зонах (для точок, які не присутні на правому зображенні, але присутні на лівому); при використанні методів для покращення невідповідності використовуються параметри, що підбираються самостійно.

1 АНАЛІЗ ДОРОЖНЬОЇ ОБСТАНОВКИ

1.1 Дорожня обстановка

Дорожня обстановка — сукупність факторів, що характеризуються дорожніми умовами, наявністю перешкод на певній ділянці дороги, інтенсивністю і рівнем організації дорожнього руху (наявність дорожньої розмітки, дорожніх знаків, дорожнього обладнання, світлофорів та їх стан), які повинен урахувати водій під час вибору швидкості, смуги руху та прийомів керування транспортним засобом [18].

Середовище та всі об'єкти, що оточують автомобіль входять до дорожньої обстановки. Тому їх аналіз являється основною задачею автомобілів без водія.

Аналіз дорожньої обстановки в сучасних автомобілях проводиться за допомогою систем визначення відстані. Ці системи називаються Lidar. Дані системи допомагають оцінювати дорожню обстановку навколо автомобіля в усі 360 градусів. Також дані системи мають високу точність оцінки відстані, приблизно ± 2 см. Але недоліком даних систем є висока ціна за дану систему, а також те, що вони можуть скоро ламатися, оскільки вони кругового принципу дії (можуть відкрутитися рано чи пізно).

Тому аналіз глибини середовища навколо автомобіля можна проводити за допомогою пари стерео відеокамер. Тобто система буде аналізувати стерео пару. Але можуть виникнути проблеми з аналізу зображень.

1.2 Проблеми аналізу зображень

Фотометричні варіації.

Фотометричні варіації – це проблеми, за яких камери по різному відображають один і той же об’єкт на фотографії. На рисунках 1.1 та 1.2, можна бачити як один об’єкт світліший за інший, або як один об’єкт засліплений сонячним промінням, в той саме час як об’єкт з іншої камери має правильну передачу кольору на фотоплівці. Фотометричні варіації виникають оскільки камери знаходяться на деякій відстані і світло падає для них під різним кутом або можуть мати різну чутливість, тому по різному передають колір.

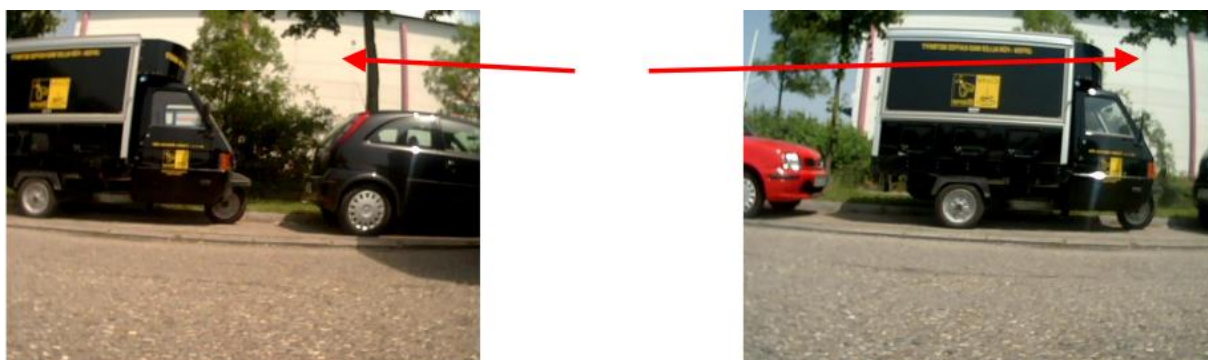


Рисунок 1.1 – Зміна кольору стіни будинку



Рисунок 1.2 – Засліплення автомобіля на лівій камері

Шум зображення датчика.

За прийом світла в камері відповідає датчик прийняття світла. Даний датчик може бути зашумленим (наприклад через недостатнє освітлення). Тому відображення фотографії з такого датчика буде низької якості, як наведено на рисунку 1.3, і буде містити шум, який буде заважати правильно аналізувати зображення.

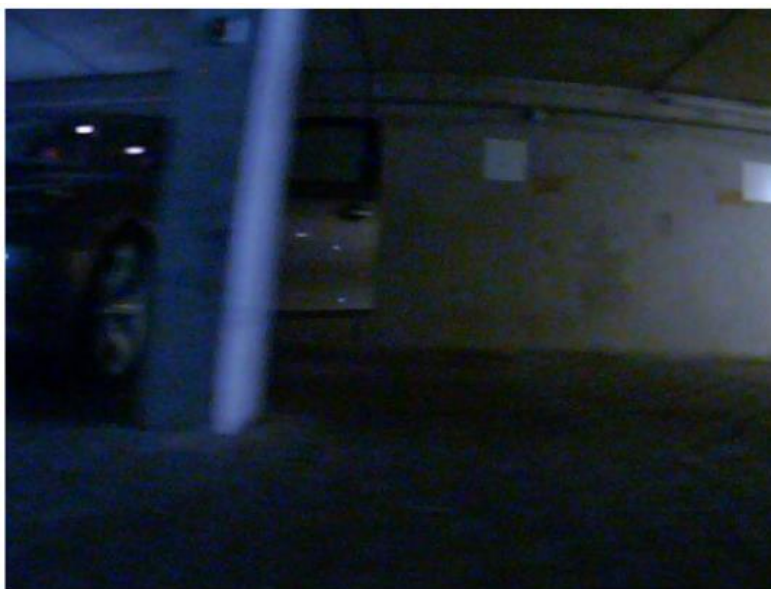


Рисунок 1.3 – Шум на зображенні лівої камери

Особливості форми об'єкту

В даному випадку в залежності від форми об'єкту, проміння може по різному падати на камеру. Для однієї камери зображення буде на певних ділянках об'єкту засліплене, в той час як на іншій камері зображення матиме природній характер. Форми об'єкту зображено на рисунках 1.4 та 1.5 (кут машини).



Рисунок 1.4 – Кут машини



Рисунок 1.5 – Бічна частина машини

Обмеження за ракурсом та поглинання.

Так як камери знаходяться на деякій відстані одна від одної, то можлива ситуація, коли один об'єкт на одній з камер буде видний повністю, а на іншій камері той же об'єкт буде зображений частково, оскільки матиме менший кут огляду об'єкту. Ситуація, коли об'єкт повністю зник на іншій стерео камері, називається поглинанням. Приклад часткового поглинання можна бачити на рисунку 1.6.



Рисунок 1.6 – Поглинання бордюру

Лінзові спотворення.

Камера з широким кутом зору має спотворення зображення і в залежності від того яке це спотворення (увігнуте або випукле) камери бувають пінхольні або фішальні [19]. Спостерігати дане спотворення можна, коли сфотографувати рівну лінію з зовнішнього світу. У фотокамері воно буде мати вигляд кривої лінії. Даний вид проблем можна бачити на рисунках 1.7, 1.8. Тому якщо камери не відкалібрувати або відкалібрувати з великою помилкою, то зробити зіставлення пікселів буде дуже важко.



Рисунок 1.7 – Радіальні спотворення лінії

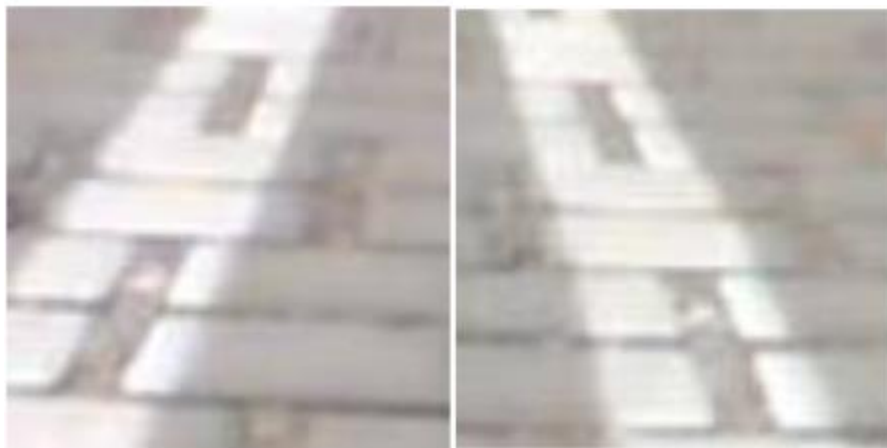


Рисунок 1.8 – Радіальні спотворення у збільшеному вигляді

Регіони з однаковою текстурою.

В комп'ютерному баченні має місце проблема, коли велика частина зображення (регіон) не має відокремлюваних ознак, тобто рисунок може бути повністю мати монотонний колір (константа). Наприклад, небо, стіна однакового кольору або окремий випадок наведений на рисунку 1.9. Тому якщо локально шукати відповідність, то відповідність знайти не вдасться.

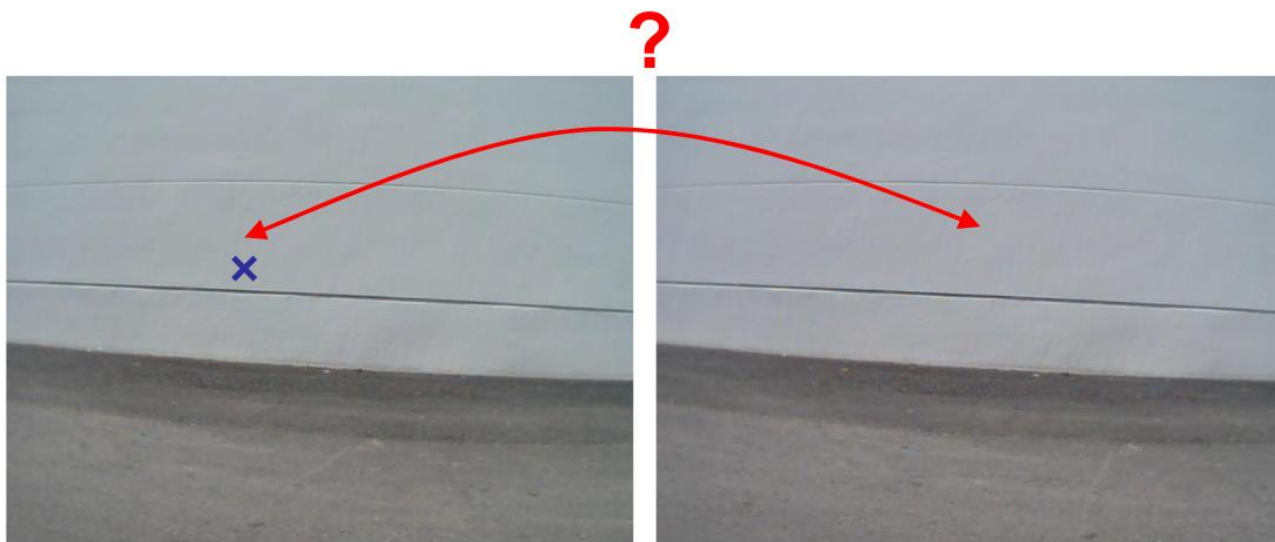


Рисунок 1.9 – Однакова текстура стіни (монотонний білий колір)

Текстури, що повторюються.

Великою проблемою зіставлення є структури, що повторюються. Це може бути наприклад паркан, різні форми архітектури або окремий випадок приведений на рисунку 1.1. На даному рисунку, крім текстур, що повторюються, ще й великі регіони з однаковою текстурою. Також дані регіони важко співставляти, оскільки невідомо який регіон потрібно обрати з деякої множини регіонів.

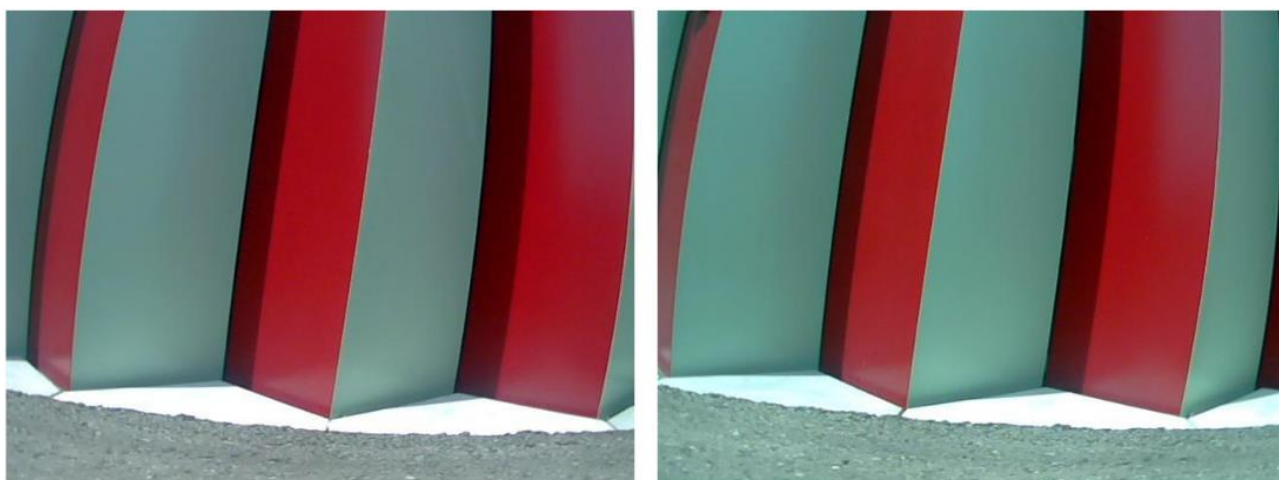


Рисунок 1.10 – Стіна

Віддзеркалення.

Однією з найбільшій проблем, це коли на одному об'єкті ми бачимо зображення іншого об'єкта через те, що поверхня об'єкта – дзеркальна. На таких об'єктах важко знайти співставлення, оскільки ми бачимо інші об'єкти. І якщо бачачи цілісний об'єкт людина може відрізнити місця з віддзеркаленням, то машині, як розуміє тільки значення пікселя зробити це дуже важко. Приклади віддзеркалення можна бачити на рисунках 1.11, 1.12.



Рисунок 1.11 – Віддзеркалення на стіні

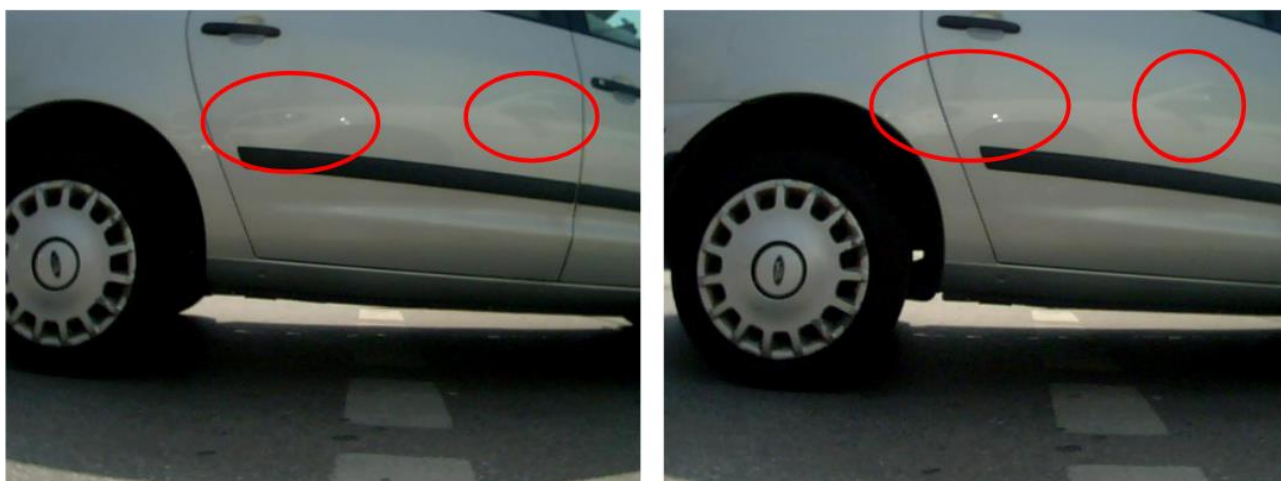


Рисунок 1.12 – Віддзеркалення на машині

Прозорість

Окремим випадком віддзеркалення є прозорість, коли бачення головного предмета перешкоджає прозора поверхня або напівпрозора (додає нечіткість бачення). Складність полягає в тому, що головний об'єкт бачимо не чітко, також перешкоджає ще те, що на прозору поверхню накладається зображення (оскільки вона має дзеркальні властивості). На рисунку 1.13 наведено приклад прозорих (скляних) об'єктів.



Рисунок 1.13 – Напівпрозорі вікна офісу

Загородження.

Загородження (Occlusion) - це проблема, коли на одній камері видно об'єкт, на іншій - ні, оскільки загородження перешкоджає бачити об'єкт. Загородження буває двох видів: зіставлення справа наліво (рисунок 1.14) та зіставлення зліва на право (рисунок 1.15).

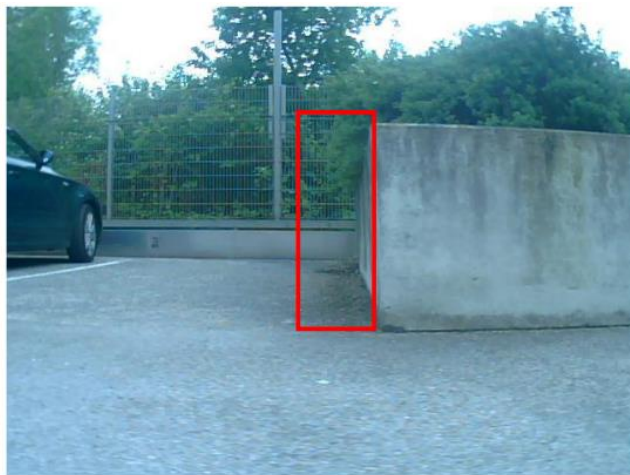


Рисунок 1.14 – Загородження (зіставлення справа наліво)



Рисунок 1.15 – Загородження (зіставлення зліва направо)

Висновки до розділу

Задача аналізу зображень з камери автомобіля є складною. Проблеми можуть бути з низькою якістю камери (шум датчика), з погодними умовами на дорозі, з текстурою об'єктів або з властивостями оболонки об'єктів дорожньої обстановки.

2 МЕТОД ФОТОГРАММЕТРІЇ

Свій початок фотограмметрія бере, безумовно, з появи фотографії і майже відразу ж стереофотографії. Розвиток техніки, літальних апаратів, оптики привело до величезного розвитку фотограмметрії і використання її в картографуванні. Значні успіхи в розвитку наук і нових галузей: радіоелектроніки, автоматичних систем управління, математичного апарату і обчислювальної техніки, фізичних відкриттів і приладобудування, космічної геодезії та картографії, комп'ютерних технологій інформаційних систем сприяли сучасному рівню і станом фотограмметрії зараз. Виникають способи аерофотограмметричних, морських, космічних зйомок з різними методами отримання інформації про поверхні земної та водної поверхні.

2.1 Основи фотограмметрії

2.1.1 Предмет вивчення фотограмметрії

Вимірювання по світловим записів це, мабуть, найкоротший і точне визначення фотограмметрії, що відбувається з грецького перекладу цього складного слова. Більш детальне академічне пояснення можна сформулювати в наступному вигляді. Наука, що дозволяє за допомогою фотографування, способів обробки знімків і спеціальних технологій отримувати зображення і визначати по ним просторове положення фізичних об'єктів на місцевості і їх характеристики має назву фотограмметрія. Крім цього фотограмметрію можна вважати новою технологією дистанційного зондування при визначенні геометричних властивостей предметів, процесів, їх аналізу та надання в

графічному вигляді відомостей по групі фотознімків, знятих з різних положень фотокамери.

Основою методу фотограмметрії є фотознімки. Вивчення і вимірювання геометрії фізичного зображення знімків надає цій основі рівень наукового підходу та практичного застосування.

Використання для вивчення об'єктів і їх кількісних характеристик і властивостей одиночних фотознімків є фотограмметричним способом. Застосування для цього двох знімків вже вважається стереофотограмметричним методом.

В сучасних умовах фотограмметричні методи застосовуються в трьох напрямках по розвитку:

- методів картографування;
- космічної фотограмметрії і технологій з отримання супутникової фотографічної інформації з об'єктів земної і планетних поверхонь з космосу;
- прикладної фотограмметрії і спеціального застосування в різних галузях суспільного, виробничої та медичної діяльності.

2.1.2 Завдання фотограмметрії

Основними завданнями такої науково-технічної дисципліни, як фотограмметрія вважаються:

- методів картографування;
- вивчення геометричних властивостей, зображених на фотографічних знімках, всіляких об'єктів місцевості, що знімається поверхні;
- розтин всіляких аналітичних зв'язків між точками місцевості і знімків;
- встановлення розбіжностей зв'язків і виключення їх причин;

- підготовка і реалізація, розроблених фотограмметричних технологій по перетворенням зображень;
- отримання планових і картографічних відображень і картини місцевості;
- постійний моніторинг, регулярні спостереження і оновлення яких завгодно змін ситуації на поверхні місцевості;
- отримання різного характеру оперативної інформації.

2.1.3 Класифікація фотограмметрії

Фотограмметрію можна розділити на дві частини фотограмметричної науки:

- загальну;
- прикладну;

До складу загальної фотограмметрії входять всі теоретичні основи, вивчення методів і засоби застосування, класичної тривимірної фотограмметрії і динамічної чотиривимірної, з урахуванням як четвертої координати часу.

До прикладної відносяться всі галузі використання фотограмметрії:

- геодезична;
- науково-технічна;
- топографічна;
- космічна;
- підводний;
- застосування при використанні природних ресурсів;
- в гірничій справі і промисловості;
- в географічних і геофізичних дослідженнях;

- в інженерної фотограмметрії при вишукуваннях, проектуванні, будівництві, експлуатації інженерних споруд;
- в археологічних розкопках і дослідженнях;
- в медичних дослідженнях і діагностуванні;
- автомобілях.

Окремим випадком фотограмметрії, що відбувається на основі стереозображення (зображень, що отримані з більше, ніж однієї позиції) і передбачає визначення трьохвимірних координат об'єкта називається стереофотограмметрія. На зображеннях визначаються спільні точки.

2.2 Систематика стерео відповідності

Більшість стерео алгоритмів складаються з наступних кроків:

- а) попередня обробка;
- б) підрахунок функції вартості;
- в) підрахунок сукупної функції вартості;
- г) оптимізація;
- д) підрахунок невідповідності;
- е) покращення невідповідності.

Існує два класи алгоритмів:

а) локальні методи: не виконують глобальної оптимізації і використовують лише простий пошук для кожного пікселя зображення.

б) глобальні методи: мінімізують глобальну енергетичну функцію і не використовують підрахунок сукупної функції вартості.

2.2.1 Попередня обробка

Для того щоб зменшити шум датчика та фотограмметричні спотворення, зазвичай використовують наступні методи:

1. Фільтр Лапласіан Гаусіан (LoG) [20]
2. Вирівнювання гістограми (приклад наведений на рисунку 2.1).



Рисунок 2.1 – Вирівнювання гістограми

Віднімання середнього значення обчисленого у сусідів для кожного пікселя. [21].

Білатеральний фільтр (приклад наведений на рисунку 2.2) [22]



Рисунок 2.2 – Білатеральний фільтр

2.2.2 Функція вартості

Вартість невідповідності оцінює схожість пікселів і є функцією від (x, y, d) .

Прості приклади функції вартості:

- Модуль різниці:
- Квадрат різниці:

Підрахунок невідповідності в локальних методах

Відповідний піксель на правому зображенні стерео пари обирається так, що схожість між пікселями максимальна (або відмінність мінімальна).

Найпростішим алгоритмом є Переможець Забирає Все (“Winner Takes All”): для кожного пікселя обираємо відповідність з найменшою функцією вартості. На рисунку 2.3 зображено, де в якості функції вартості обрано модуль різниці.

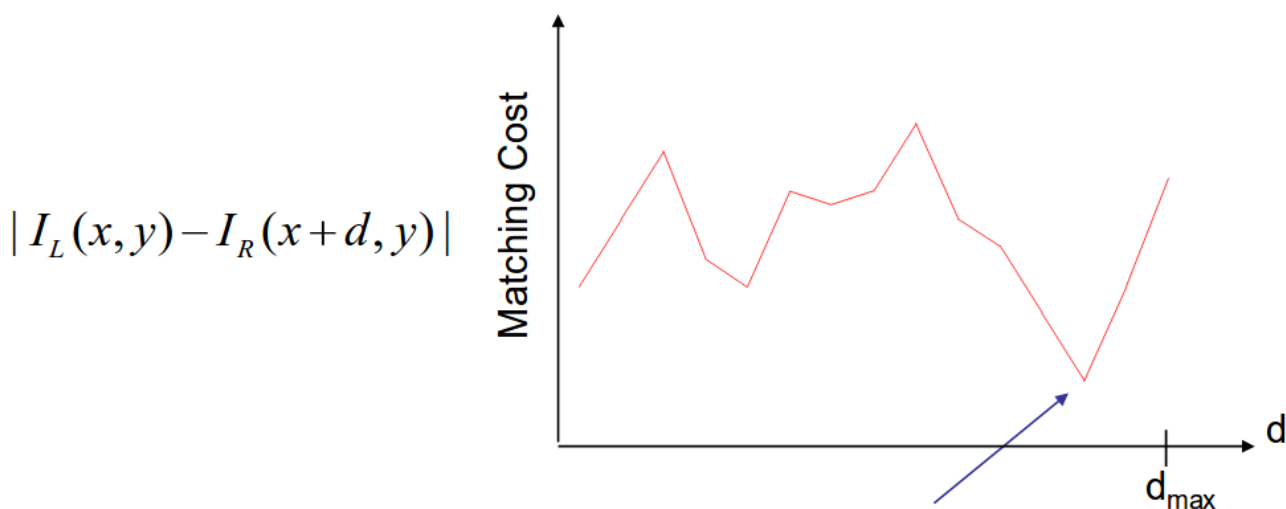


Рисунок 2.3 – Переможець забирає все

Отримана карта глибини дуже зашумлена (рисунки 2.4), оскільки висока невизначеність в текстурних регіонах.

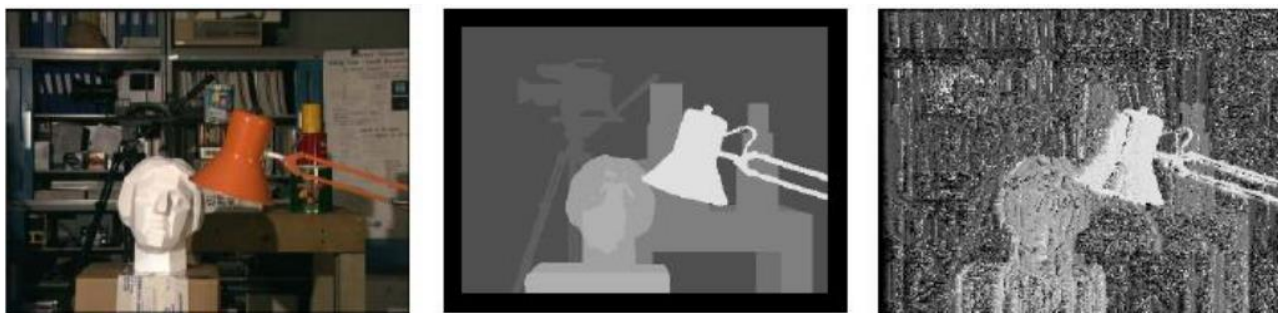


Рисунок 2.4 – Зліва на право: Ліве зображення, правильна карта глибини, карта глибини обчислена за допомогою WTA

Тому наступним кроком було використання сукупної функції вартості: порівнювати не окремі пікселі, а групу пікселів.

2.2.3 Сукупна функція вартості

Порівнювати замість пікселів, вікно пікселів, центр якого цікавий нам піксель. Вікно пікселів наведено на рисунку 2.5 (в червоному квадратику).



Рисунок 2.5 – Вікно пікселів навколо головного пікселя

Основні приклади сукупної функції вартості:

- Сума абсолютних різниць:
- Сума квадратичних різниць:
- Нормалізована крос-кореляція.
- Взаємна інформація.

Розглянемо приклад, в якому будемо використовувати стратегію «Переможець забирає все», але на основі алгоритму сукупної суми квадратичних різниць. Результат наведений нижче на рисунку 2.6 .

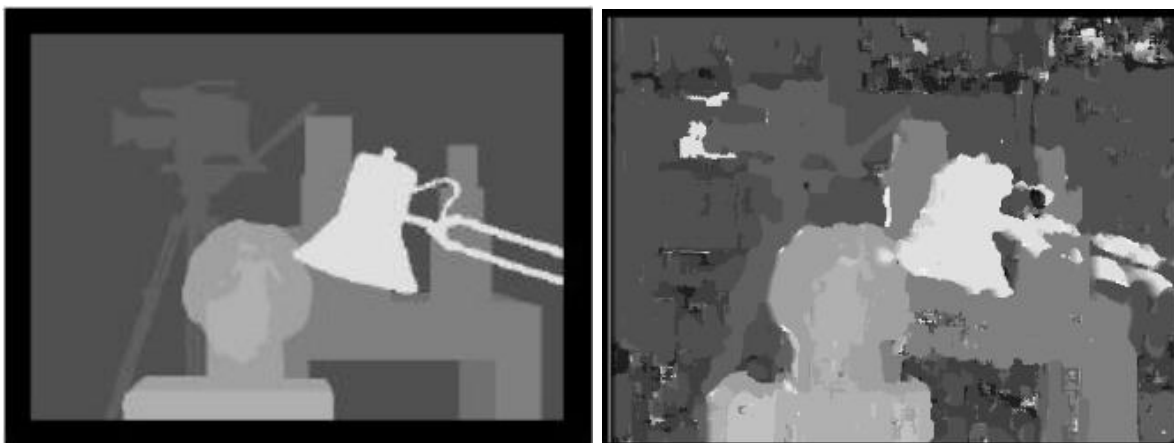


Рисунок 2.6 – Сукупна сума квадратичних різниць. Зліва направо: правильне зображення; отриманий результат

В даному випадку бачимо що ще багато помилок.

Особливості сукупної функції вартості:

- припускається, що в межах вікна глибина константа. Дане припущення порушується, в розривах глибини, а також в нахилених і не плоских поверхнях;
- погано працює з повторюваними текстурами;
- підрахунок відповідності для одноманітних поверхонь все ще невизначене;
- структуру може бути втрачено, якщо предмет менше, ніж вікно.

Приклади помилок наведені на рисунку 2.7 .

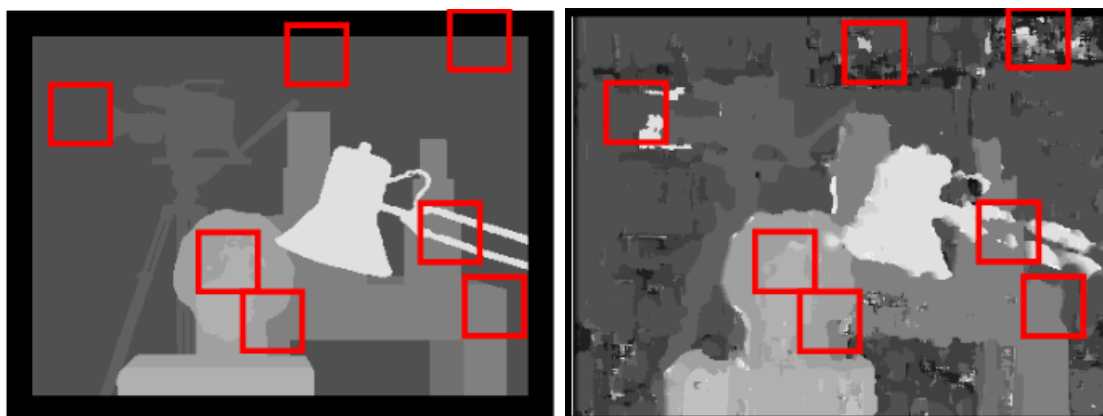


Рисунок 2.7 – Помилки сукупної функції вартості

Проблеми з похилими поверхнями.

Неявно передбачається, що коли ми обробляємо зображення, поверхня в вікні, яким ми проходимо по картинці є плоско-фронтальними. Тобто, коли функція зміни на площині YZ є похилою лінією, що зображено на рисунку 2.8 , то потрібно використовувати і похиле ковзке вікно.

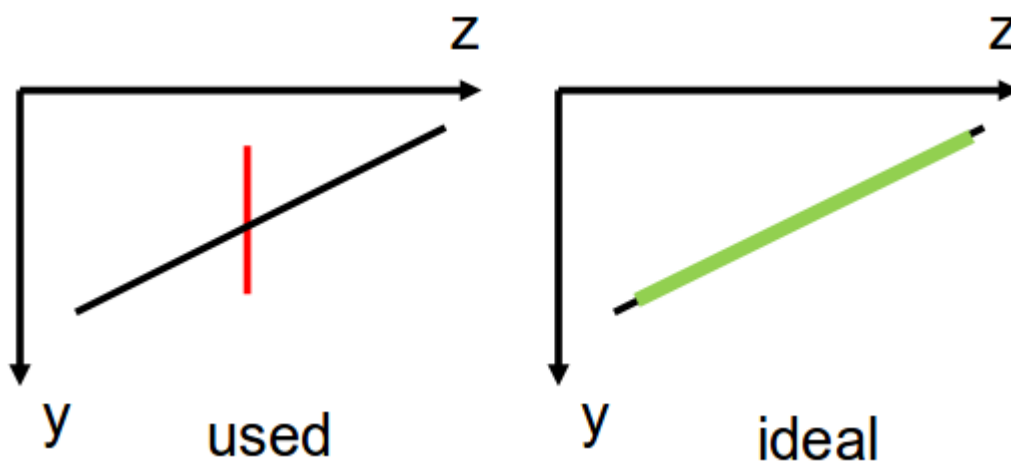


Рисунок 2.8 – Залежність зміни Y координати від глибини

Приклад наведено на рисунку 2.9 .Дана ситуація зустрічається дуже часто, особливо в середовищі навколо автомобіля. Похилим(зел. колір) і прямим(черв. колір) нарисовано ковзке вікно.

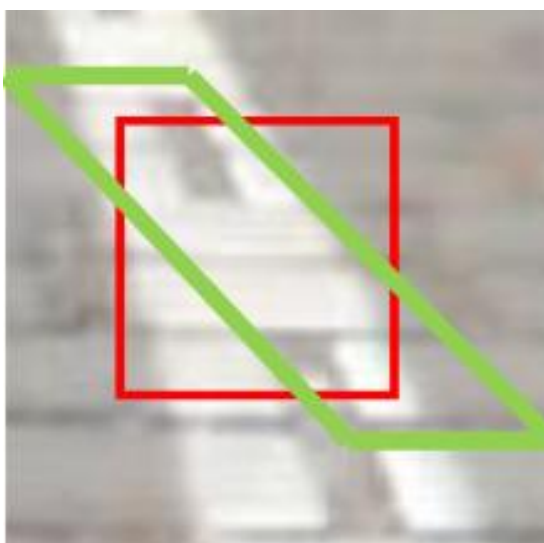


Рисунок 2.9 – Похила поверхні з ковзким вікном

В ідеалі, ковзке вікно має враховувати ці спотворення. Але це проблема, тому що, щоб правильно вибрати кут ковзкого вікна необхідно знати зміну глибини на зображенні, а глибина є невідомою, тобто саме те, що ми шукаємо.

Проблеми з розривами.

Проблема з розривами, полягає в тому, що відповідні вікна захоплюють пікселі, що знаходяться на різних глибинах. Приклад наведено на рисунку 2.10 та 2.11 .



Рисунок 2.10 – Ковзке вікно з розривами глибини



Рисунок 2.11 – Детальне зображення розриву в межах вікна

В зеленими лініями з рисунку 2.11 показано межу розриву. Червоними лініями вказано невідповідність об'єктів в ковзкому вікні. Тобто об'єкти з більшою глибиною, можуть бути зміщені в стерео-парі. Це призводить до неправильно підрахованої функції вартості. Закритий (англ. Occluded) піксель не може бути порівняний з будь-яким іншим пікселем сусідньої стерео-пари.

Можливі методи усунення: багаторазові вікна (розділити ковзке вікно на більшу кількість вікон) [23] та пересувні вікна (змінювати центр вікна динамічно).

Багаторазові та пересувні вікна лише частково дають покращення. Тому кращим виходом є використання попередньої сегментації та додавання інформації про кольори до ковзкого вікна.

Тому можна використовувати зважену функцію вартості.

Для кожного пікселя у ковзкому вікні, підраховуємо вагу, яка ґрунтується на схожості кольору: де

це функція, що підраховує вагу для пікселя у ковзкому вікні з центром в

Тому відповідні пікселі, що мають схожий колір, отримують високий ваговий коефіцієнт і подібність може бути підрахована більш точно.

Популярні методи для вибору вагової функції: Adaptive Support Weight [24] та Geodesic Support Weights [25].

Приклад роботи алгоритму Geodesic Support Weight зображена на рисунку 2.12



Рисунок 2.12 – Результат зваженої вагової функції при розриві

Як бачимо з рисунка 2.12 що результат став помітно кращий на розривах, єдине що є розмитість глибини на контурі розриву.

Повний результат роботи зваженого алгоритму Geodesic Support Weight зображено на рисунку 2.13.

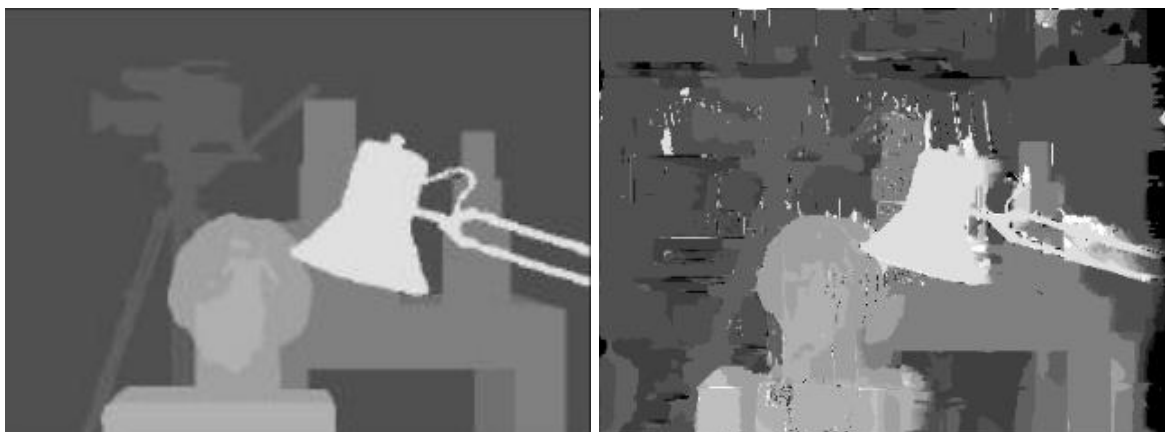


Рисунок 2.13 – Повний результат роботи алгоритму Geodesic Support Weight

Проблеми з однорідними та повторювальними поверхнями

Прикладом однорідної поверхні можна бачити на рисунку 2.14 , такожи детальніше можна наглядати на рисунку 2.15 .

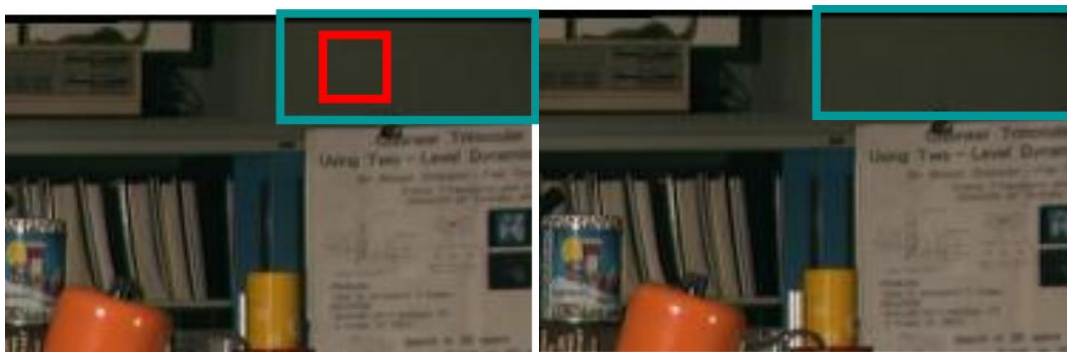


Рисунок 2.14 – Ковзке вікно і однорідна поверхня

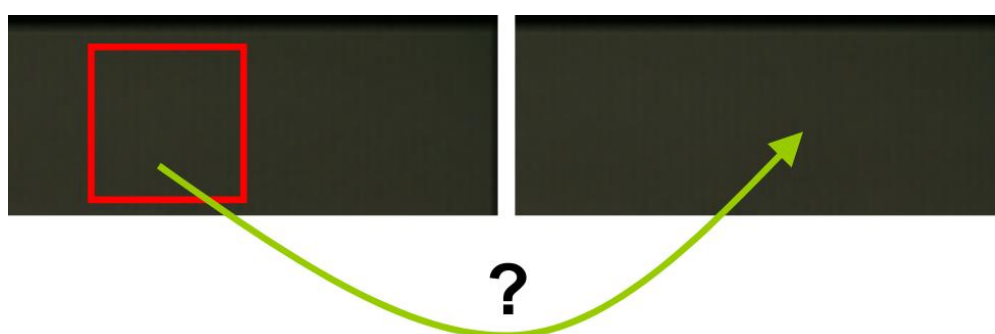


Рисунок 2.15 – Невідома відповідність ковзкого вікна в збільшеному вигляді

В даному випадку, як бачимо з рисунків, при русі ковзкого вікна виникає багато значень сукупної функції вартості, які дуже схожі одна на одну. Тому якщо в нас сенсор чутливий до шуму, то це призведе до неправильної відповідності лівого і правого пікселя.

Можливі шляхи покращення: використання мінливих вікон, тобто в залежності від ситуації, коли в нас багато схожих значень, використовувати вікно більшого розміру [26]; іншим шляхом є вимірювання важливості кожного мінімуму і відкинути інші мінімуми.

Висновок локальних методів

Однак, незважаючи на велику кількість недоліків методів, що базується на ковзкому вікні, дані методи можуть використовуватися на практиці, оскільки мають наступні переваги:

- простота (зрозумілість алгоритму та можливість легко запрограмувати);

- швидкість (можуть швидко виконуватися на більшості апаратного забезпечення);
- пам'ять (потребують малої кількості пам'яті);
- стійкість (відносно точний до основних текстур на зображеннях стереопари);
- кількість пам'яті необхідної для виконання є відносно малою, оскільки жодної додаткової інформації не потрібно, окрім значення самого пікселя.

Також може бути використаний невеликий допоміжний засіб з динамічного програмування – вираховувати крок за кроком наступне вікно знаючи значення попереднього. Тобто знаючи значення вікна в точці , значення в положенні та в може бути обраховане без перерахування всіх координат у вікні, а лише відніманням та додаванням рядків та стовпців відповідно. Приклад наведено на рисунку 2.16 .

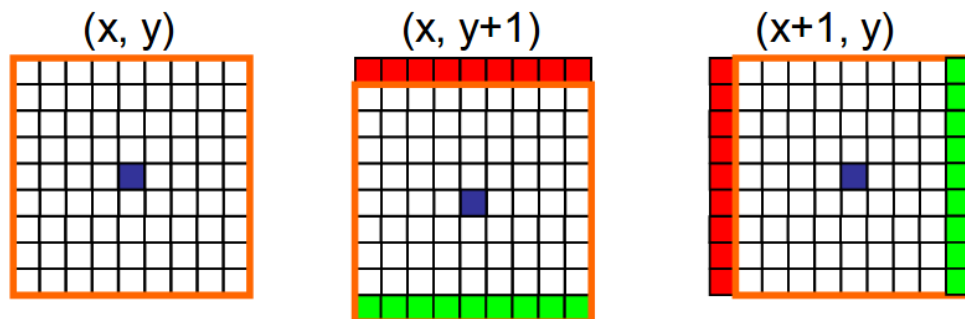


Рисунок 2.16 – Обрахунок сусіднього вікна, знаючи попереднє

Можливістю покращення швидкості даних методів є використання принципу SIMD (одна інструкція, багато даних). Використання даного принципу вже вбудовано в багато процесорів, а також написано багато бібліотек з використання даного принципу . З використанням даного принципу можна опрацьовувати дані паралельно, в даному випадку опрацьовувати більшу кількість пікселів одночасно.

Також можливістю швидкого обчислення може бути використання SSE2 інструкцій.

Щоб пришвидшити швидкістю, можна обраховувати значення різниць не в форматі плаваючої точки, а в цілих числах.

З використанням даних технік, можна отримати, що час обробки зображення розміром 640x400 з максимальною невідповідністю 40, може бути обраховано за час всього 60 мс.

Також можливим покращенням може бути використання даного методу [27]. Це дає можливість не обробляти всі можливі значення невідповідності, а зупинятися на відповідних точках при виконанні особливих умов. Це дає можливість пришвидшити швидкість обробки однієї картини десь на 30%.

Глобальні та неглобальні методи оптимізації.

Використання локальних методів оснований на кожному вікні є недостатньо стійкою групою методів, оскільки обчислення невідповідності відбувається для кожного пікселя окремо.

Проблеми локальних методів:

- сукупна функція вартості, яка використовується для надійності методу, вносить помилки при розриві глибини та похилих поверхнях;
- підрахунок невідповідності для однорідних областей та повторювальних текстур залишається проблематичним.

Для виправлення заданих проблеми спробуємо використовувати глобальні методи оптимізації. Дана стерео проблема формулюється як мінімізації енергії (формула 2.1):

(2.1)

де α – вимірює схожість пікселів, а β – штрафує варіації невідповідності, D – карта невідповідності, також функція, що переводить з простору зображення в простір невідповідності.

Значення даних α можуть бути взяті з абсолютних різниць.

Значення гладкості – це бінарні значення і можуть бути обраховані наступним чином (формула 2.2 та 2.3):

(2.2)

де

(2.3)

Загальновідомі методи для вирішення даної проблеми є наступні методи: Динамічне програмування, Scanline Optimization, Graph Cuts, Belief Propagation та інші.

Спочатку в глобальних методах будується граф на все зображення, де кожен піксель є вершиною і має чотирьох сусідів, наприклад рисунок 2.17 .

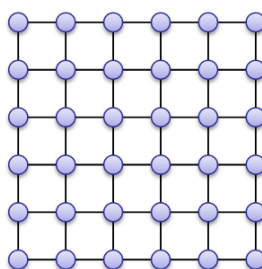


Рисунок 2.17 – Створення графу з малюнку

Необхідно для кожної вершини (пікселя) встановити невідповідність так, щоб це було узгоджено з значенням даних (інтенсивністю зображення) та щоб значення невідповідності не сильно відрізнялося від локальній області (значення гладкості), що задається обмеженням на граф.

В загальному випадку, це NP-складна задача. Але якщо використовувати Динамічне програмування, то дану задачу можна вирішити за поліноміальний час.

Результат використання методу Динамічного програмування наведений на рис. 2.18 ; методу Belief Propagation на рис. 2.19 ; методу Graph Cuts – на рис. 2.20



Рисунок 2.18 – Невідповідність методом Динамічного Програмування



Рисунок 2.19 – Невідповідність методом Belief Propagation



Рисунок 2.20 – Невідповідність методом Graph Cuts

Однак за якість необхідно платити. В даному випадку часом обробки зображення. Підрахунок невідповідності для одного зображення кожним з цих методів займає більше, ніж 30 с.

2.2.4 Покращення невідповідності

До цього часу ми розглядали лише обрахунок невідповідності в цілих значеннях. Але насправді невідповідність – це неперервна величина. Тому справжнє значення невідповідності може бути наближено використовуючи параболу (див. рис. 2.21) або інтерполяції різних типів (лінійна, кубічна).

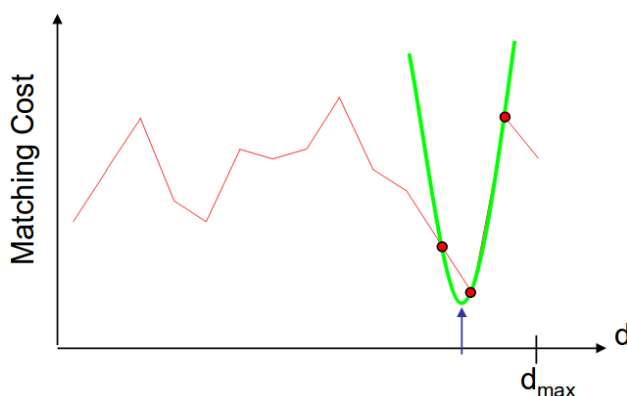


Рисунок 2.21 – Наближене значення мінімуму параболою

Ще одним допоміжним засобом для покращення невідповідності є перевірка Left-Right Consistency Check. Дана перевірка дозволяє виявити недійсні значення невідповідності.

Алгоритм виявлення недійсних значень:

- а) визначити невідповідність для кожного пікселя лівого зображення (зліва-направо);
- б) визначити невідповідність для кожного пікселя правого зображення (справа-наліво);
- в) якщо невідповідність (піксель) не співпадає між лівою та правою картою невідповідності, то даний піксель вважається недійсним. Приклад наведено на рис. 2.22 .

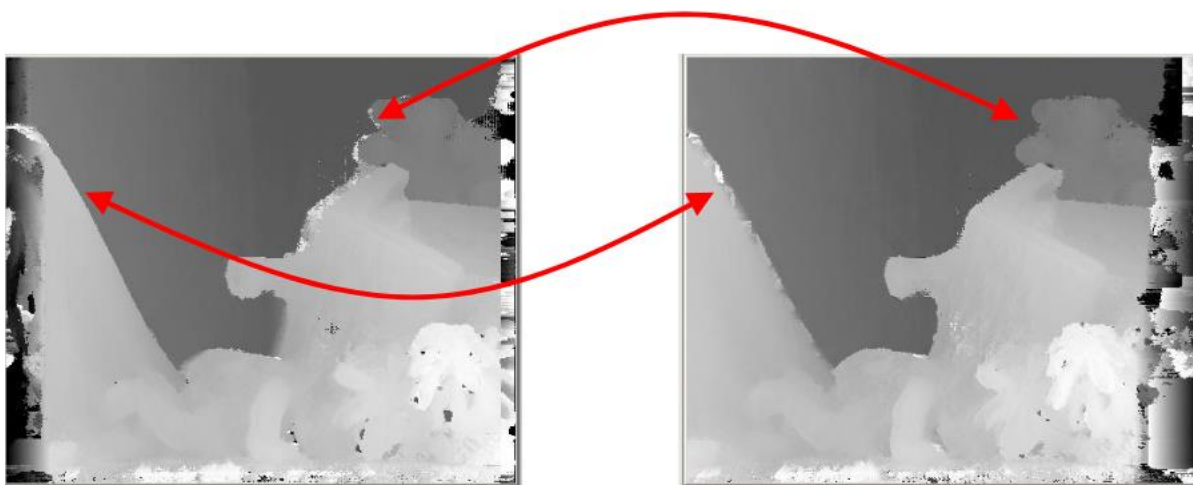


Рисунок 2.22 – Виявлення недійсного пікселя невідповідності

2.3 Огляд літератури

На сьогоднішній день існує велика кількість методів та книжок для оцінки невідповідності зі стерео пари. Ми оглянемо лише декілька з них і зробимо наголос на методах, які використовують згорткові нейронні мережі.

Одними з основних методів для побудови карти глибини зі стерео пари в комп'ютерному баченні є GBM і SGBM. Недоліком цих методів є те, що вони не можуть аналізувати текстуру зображення, не може аналізувати дзеркальні поверхні, погано справляються з закритими областями (occlusion) тощо. На Kitti2015 [17] змаганні дані алгоритми мають занадто високу помилку та тривалий час роботи.

На відміну від типових алгоритмів [16, 28], які виконуються в чотири кроки (див. вище), методи основані на згорткових нейронних мережах оцінюють невідповідність безвідносно до цих кроків. Умовно підходи ЗНМ можна поділити на дві категорії: навчання функції вартості та навчання кінець-в-кінець.

2.3.1 Навчання функції вартості.

Одними з основних методів для побудови карти глибини зі стерео пари в комп'ютерному баченні є GBM і SGBM. Недоліком цих методів є те, що вони не можуть аналізувати текстуру зображення, не може аналізувати дзеркальні поверхні, погано справляються з закритими областями (occlusion) тощо. На Kitti2015 [17] змаганні дані алгоритми мають занадто високу помилку та тривалий час роботи.

На відміну від звичайних метрик для оцінки функції вартості, такі як сума відмінностей абсолютної інтенсивності (SAD), нормалізована перехресна

кореляція (NCC), згорткові нейронні мережі використовують для вимірювання схожості двох патчів зображення. Han та ін. [29] надали мережу: MatchNet – виділяє ознаки з двох патчів і на основі повнозв’язних шарів робить висновок про їх схожість. Даний метод досяг найкращої точності на Kitti2015 [17] і Middlebury [14] стерео датасетах в той час. Даний метод показав, що порівнювати значення пікселів у функції вартості було недостатньо, в той час коли згорткова нейронна мережа може навчитися більш стійко знаходити невідповідність, використовуючи карти ознак зображення. Але в даній роботі були недоліки, а саме тривалий час підрахування невідповідності ($> 60s$); тому даний метод не можна використовувати для отримання невідповідності в закритих зонах (для точок, які не присутні на правому зображенні, але присутні на лівому); при використанні методів для покращення невідповідності використовуються параметри, що підбираються самостійно. На відміну від даного підходу, Zbontar і Lecun [11] за допомогою ЗНМ бінарно класифікували патчі зображень розмірів 9×9 . Також в даній роботі використовувались методи для пост-обробки: перехресна агрегація, напів-глобальне зіставлення (SGM), перевірка узгодженості зліва направо, по-піксельне покращення та медіанний і білатеральний фільтри. Дана архітектура потребує багато повторів запуску для підрахунку функції вартості (залежить від максимальної величини невідповідності). Park і Lee [30] запропонували по-піксельний пірамідний пулінг для збільшення рецепторного поля під час порівняння двох патчів. На відміну від бінарної класифікації між двома патчами, Luo [31] запропонували вчити розподіл ймовірності для кожного можливого значення невідповідності. Хоча використання згорткових нейронних мереж значно покращило порівняння патчів, але після цього ідуть кроки вартісної агрегації, оптимізація невідповідності та покращення невідповідності, що загалом потребують покращення, як в точності, так і в часі обчислень.

2.3.2 Навчання кінець-в-кінець

В цій категорії, ЗНМ повністю оцінює невідповідність, що дозволяє тренувати мережу з вчителем кінець-в-кінець, подаючи на вхід стерео пару і правильне зображення невідповідності. Mayer [15] першим запропонував кінець-в-кінець підхід з енкодер-декодер архітектурою (Dispnet) для оцінки невідповідності. Підрахунок функції вартості включено в енкодер частині архітектури. Одними з основних методів для побудови карти глибини зі стерео пари в комп'ютерному баченні є GBM і SGBM. Недоліком цих методів є те, що вони не можуть аналізувати текстуру зображення, не може аналізувати дзеркальні поверхні, погано справляються з закритими областями (occlusion) тощо. На Kitti2015 [17] змаганні дані алгоритми мають занадто високу помилку та тривалий час роботи.

Схожі архітектури на Dispnet були застосовані для оцінки потоків (optical flow) в роботах [32, 13]. Кендал [12] з архітектурою мережі GC-Net, використовує 3D згортки після підрахунку функції вартості. Особливістю даних архітектур є те, що вони мають малий час виконання (менше 1 с.). Також в деяких роботах [33, 34] для оцінки глибини з одного зображення, а не стерео пари використовуються згорткові мережі, але точність набагато гірша, ніж в стерео парі.

До цієї ж категорії належить і наш метод. В нашій роботі ми змінили архітектуру однієї варіації архітектури Dispnet, так щоб отримати невідповідність високої якості. Завдяки цьому ми змогли досягти великої точності для оцінки невідповідності за цілком прийнятний час.

Висновки до розділу

Фотограмметрія – важлива наука, що аналізує фотографії. Окремим випадком фотограмметрії є стереофотограмметрія. Основні етапи обробки зображення за допомогою стереофотограмметрії для отримання глибини: попередня обробка, визначення функції вартості, підрахунок сукупної функції вартості та покращення невідповідності.

3 ПОБУДОВА МОДЕЛІ ЗА ДОПОМОГОЮ МЕТОДУ ФОТОГРАММЕТРІЇ

На сьогодні згорткові нейронні мережі (ЗНМ) успішно застосовуються у багатьох областях комп'ютерного зору, особливо для обчислення карти глибини зі стерео пари. Однак проблема визначення високої якості карти глибини залишається ще не вирішеною. Ми пропонуємо архітектуру згорткової нейронної мережі, яка швидко дозволяє отримати карту глибини з великою точністю. Дана мережа будується на типі енкодер-декодер, з кореляційним шаром. Експеримент показав, що нам вдалося покращити точність обчислення карти глибини на Kitti онлайн змаганні на пів відсотка, при цьому час обчислення карти глибини становить всього 68мс.

3.1 Згорткові нейронні мережі

Швидкий темп розвитку комп'ютерних засобів спричинив широке застосування систем штучного інтелекту в різних галузях науки і техніки. Для створення штучного комп'ютеру потрібно опрацьовувати п'ять каналів обробки інформації, а саме зорового, слухового, нюхового, смакового, тактильного. Велике значення в штучному інтелекті приділяється обробці візуальної інформації. Це зумовлено великим розповсюдженням зорових образів. Обробка візуальної інформації надзвичайно складна задача, оскільки інформація може бути надлишковою і складною для аналізу.

На сьогоднішній день для обробки візуальної інформації застосовують системи комп'ютерного бачення, в яких основною задачею є розпізнавання образів.

Виявлення об'єктів вважається найголовнішим завданням комп'ютерного зору. Решта інших розробок в області комп'ютерного зору досягається шляхом невеликих удосконалень.

У реальному житті, кожен раз, коли люди відкриває свої очі, вона несвідомо виявляє об'єкти, класифікує їх.

Задачі розпізнавання зображень та класифікації об'єктів на сьогоднішній день є актуальними і виникають практично в усіх сферах людської діяльності:

- соціальні мережі (напр. розпізнавання обличчя);
- системи організації інформації (напр. індексація баз даних зображень);
- кримінальна сфера (напр. класифікація доказів, розпізнавання обличчя, відбитку пальця);
- військова сфера (напр. класифікація об'єктів для визначення ступеня загрози);
- комп'ютерне бачення (Computer vision) (розпізнавання та класифікація зображень).

Розглянемо проблеми розпізнавання, з яким справляється ЗНМ:

- а) зміна кута бачення;

Один і той же об'єкт може мати різні позиції і кути в зображенні відносного положення об'єкта і спостерігача, як зображено на рисунку 3.1 Хоча об'єкт легко розпізнати, але дуже важко навчити комп'ютер розпізнавати, що це той самий об'єкт.



Рисунок 3.1 – Зображення об'єкта під різними кутами

б) зміна освітлення;

Різні зображення можуть мати різні умови освітлення (рис. 3.2).



Рисунок 3.2 – Темне зображення кішки

Хоча зображення (рис. 3.2) темне, ми все ще можемо розпізнати, що це кішка. Для комп'ютера це є однією з проблем.

в) приховані частини об'єктів;

Зображення не обов'язково повинні бути повними. Маленькі чи великі частини зображень можуть бути приховані, що робить задачу розпізнавання важкою. Наприклад рисунок 3.3 .



Рисунок 3.3 – Прихована частина цуценя

г) шум на задньому плані зображення.

Деякі зображення можуть зливатися з фоном. Наприклад рисунок 3.4 .



Рисунок 3.4 – Людина зливається з заднім фоном

Дану задача важко розв'язати за допомогою комп'ютера.

3.2 Кореляційний шар

Для того щоб оцінити невідповідність між картами ознак лівого та правого зображення ми використовували кореляційний шар [32]. Даний шар нейронної мережі здійснює мультиплікативне порівняння патчів лівої та правої карт ознак.

Нехай задано наступні дві карти ознак: I та J , де I – це відповідно ширина, висота та кількість каналів, даний кореляційний шар дозволяє порівнювати кожний патч з I з кожним патчем з J .

Зараз опишемо лише одне порівняння двох патчів. Кореляція патчу лівої карти ознак з центром в (x_1, y_1) та правої карти ознак з центром в (x_2, y_2) визначається як формула 3.1:

$$C_{ij} = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} I(x, y) \cdot J(x, y) \quad (3.1)$$

для розміру патчу (w, h) . Можна замітити, що формула 3.1 аналогічна операції згортки, де замість карти правого зображення використовуються ваги фільтру, але замість того, щоб робити згортку між фільтром, проводиться згортка з картами ознак іншого зображення. Даний шар не містить вагових коефіцієнтів, оскільки використовує лише дані лівих та правих карт ознак, і не потребує навчання.

Обчислення C_{ij} потребує $w \cdot h$ множень. Порівнюючи всі можливі варіанти, виходить необхідно зробити таких множень велику кількість: $w \cdot h \cdot w \cdot h$, що займає дуже велику кількість часу для обрахунку. Тому ми обмежили максимальну невідповідність параметром даного шару. Також як і в згортковому шарі додали stride та padding (заповнення нулями).

Маючи максимальну невідповідність Δ , для кожної координати лівої карти ознак (з урахуванням stride) ми обчислюємо кореляцію Δ тільки в сусідів розміром Δ .

Ми організували відносні зміщення в канали, тобто вихід має розміри $\frac{H}{\Delta} \times \frac{W}{\Delta}$. Також для даного шару було обраховано зворотній хід (для методу зворотного поширення помилки).

Всі обчислення були написані на мові програмування CUDA, що значно пришвидшило виконання даної операції.

3.3 Підхід

Mayer [15] розглядають декілька архітектур, а саме DispNet та DispNetCorr1D. DispNetCorr1D – це архітектура DispNet, яка має додатковий Correlation layer. Ми обрали DispNetCorr1D як базову, оскільки вона дає кращі результати, ніж DispNet [15]. Наша архітектура має енкодер-декодер тип. Однак, на відміну від DispnetCorr1D, в якому вихідна невідповідність має розширення вдвічі менше за вхідне зображення, наша мережа має ще додаткові шари, які збільшують вихідну невідповідність до вхідного розміру стерео пари. Як буде показано в секції 4, що дана мережа показала більш кращі результати на датасетах Kitti та FlyingThings3D. Нашу архітектуру ми назвали DispNetEqual (Equal – означає рівний, тобто вихідне розширення невідповідності, такого ж розміру як і вхідне розширення стерео пари).

Оцінка невідповідності відбувається в декодер частині за різного масштабу картинки, як показано в таблиці 3.1.

Таблиця 3.1 – Архітектура DispNetEqual

| Шар | К | S | Канали | Вх. | Вих. | Вхідні канали |
|--------------|---|---|-----------|-----|------|--------------------------|
| conv1_1 | 7 | 2 | 6/64 | 1 | 2 | left+right |
| conv1_2 | 3 | 1 | 64/64 | 2 | 2 | conv1_1 |
| conv2_1 | 5 | 2 | 64/128 | 2 | 4 | conv1_2 |
| conv2_2 | 3 | 1 | 128/128 | 4 | 4 | conv2_1 |
| Correlation | - | - | 128/128 | - | - | conv2_2 |
| conv3_1 | 3 | 2 | 128/256 | 4 | 8 | Correlation |
| conv3_2 | 3 | 1 | 256/256 | 8 | 8 | conv3_1 |
| conv4_1 | 3 | 2 | 256/512 | 8 | 16 | conv3_2 |
| conv4_2 | 3 | 1 | 512/512 | 16 | 16 | conv4_1 |
| conv5_1 | 3 | 2 | 512/512 | 16 | 32 | conv4_2 |
| conv5_2 | 3 | 1 | 512/512 | 32 | 32 | conv5_1 |
| conv6_1 | 3 | 2 | 512/1024 | 32 | 64 | conv5_2 |
| conv6_2 | 3 | 1 | 1024/1024 | 64 | 64 | conv6_1 |
| pr6 | 3 | 1 | 1024/1 | 64 | 64 | conv6_2 |
| deconv5 | 4 | 2 | 1024/512 | 64 | 32 | conv6_2 |
| up6_to_5 | 4 | 2 | 1/1 | 64 | 32 | pr6 |
| conv_concat5 | 3 | 1 | 1025/512 | 32 | 32 | conv5_2+deconv5+up6_to_5 |
| pr5 | 3 | 1 | 512/1 | 32 | 32 | conv_concat5 |
| deconv4 | 4 | 2 | 512/256 | 32 | 16 | conv_concat5 |
| up5_to_4 | 4 | 2 | 1/1 | 32 | 16 | pr5 |
| conv_concat4 | 3 | 1 | 769/256 | 16 | 16 | conv4_2+deconv4+up5_to_4 |

Продовження таблиці 3.1

| | | | | | | |
|--------------|---|---|---------|----|----|--------------------------|
| pr4 | 3 | 1 | 256/1 | 16 | 16 | conv_concat4 |
| deconv3 | 4 | 2 | 256/128 | 16 | 8 | conv_concat4 |
| up4_to_3 | 4 | 2 | 1/1 | 16 | 8 | pr4 |
| conv_concat3 | 3 | 1 | 385/128 | 8 | 8 | conv3_2+deconv3+up4_to_3 |
| pr3 | 3 | 1 | 128/1 | 8 | 8 | conv_concat3 |
| deconv2 | 4 | 2 | 128/64 | 8 | 4 | conv_concat3 |
| up3_to_2 | 4 | 2 | 1/1 | 8 | 4 | pr3 |
| conv_concat2 | 3 | 1 | 193/64 | 4 | 4 | conv2_2+deconv2+up3_to_2 |
| pr2 | 3 | 1 | 64/1 | 4 | 4 | conv_concat2 |
| deconv1 | 4 | 2 | 64/32 | 4 | 2 | conv_concat2 |
| up2_to_1 | 4 | 2 | 1/1 | 4 | 2 | pr2 |
| conv_concat1 | 3 | 1 | 97/32 | 2 | 2 | conv1_2+deconv1+up2_to_1 |
| pr1 | 3 | 1 | 32/1 | 2 | 2 | conv_concat1 |
| deconv0 | 4 | 2 | 32/32 | 2 | 1 | conv_concat1 |
| up1_to_0 | 4 | 2 | 1/1 | 2 | 1 | pr1 |
| conv_concat0 | 3 | 1 | 39/32 | 1 | 1 | input+deconv0+up1_to_0 |
| pr | 3 | 1 | 32/1 | 1 | 1 | conv_concat0 |

В даній мережі Correlation шар рахує функцію вартості в просторі ознак. На даному етапі використовуються неглибокі карти ознак, а саме вихід з шару “conv2_2”. Далі вихід Correlation шару об’єднується з картами ознак лівого зображення, в надії на те, що наступна частина мережі покращить функцію вартості і удосконалисть вихідну невідповідність.

Вихід мережі – це вихід з шару pr. Зменшувальна частина містить conv1_1 – conv6_2 та Correlation шари. Розширювальна частина містить згорткові шари (pr, conv_concat) та deconvolution шари (deconv, up..to..).

Також, зображення архітектури нейронної мережі, наведено на рисунку 3.5

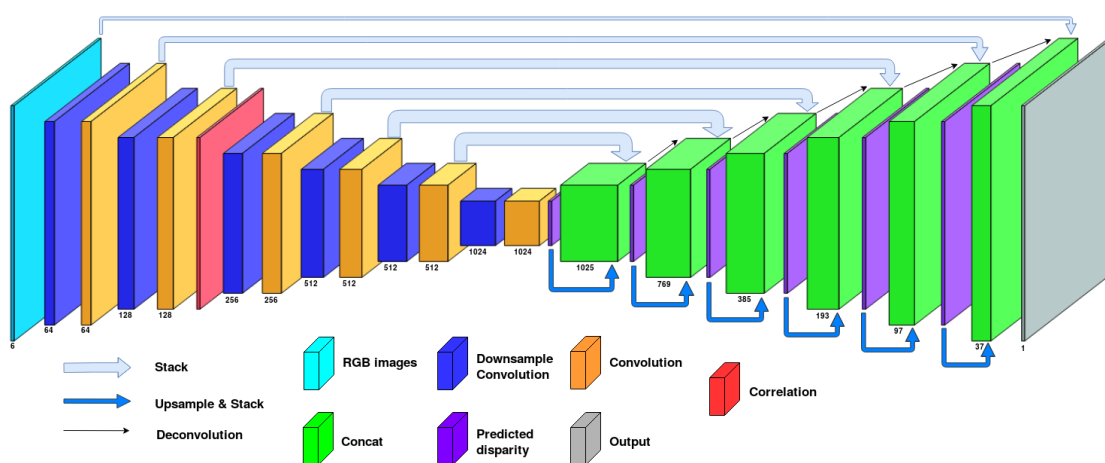


Рисунок 3.5 – Графічне зображення архітектури мережі

Окрім шарів в кінці, ми додали ще декілька шарів на початку.

3.4 Тренування

3.4.1 Датасети

Для тренування великої згорткової мережі необхідно датасет з стерео парою та правильною глибиною. Розглянемо основні датасети.

Middlebury dataset [14] містить тільки 23 стерео пари зображень з правильною глибиною. Максимальне значення невідповідності (disparity) є 10, що є малим значенням.

Kitti2015 [17] датасет є вже значно більшим і був відзнятий у реальних дорожніх ситуацій з рухомого автомобіля, одночасно знімаючи зображення з камери та лазерного сканера. Даний датасет містить 200 стерео зображень для тренування та 200 стерео зображень для оцінки онлайн. Розмір зображень 375 x 1240. Максимальне значення невідповідності сягає 60, що є вже практично використовуваним. Недоліком даного датасету є те, що значення глибини в ньому є розрідженими на 50%, для верхньої половини зображення значення глибини відсутні та через рух автомобіля значення глибини можуть бути небагато зміщеними. Але його реалістичність та достатньо невелика кількість сприяла його використанню для тренування мережі.

Syntel [35] датасет отриманий з рендерингу штучних сцен з акцентом на реалістичних ефектах та пропорціях в зображенні. Даний датасет має дві версії: звичайну та фінальну. Фінальна версія має багато ефектів, такі як туман, в той час як звичайна не містить цих ефектів. Syntel датасет – один з найбільших датасетів, в якому доступно більше тисячі зображень для тренування для кожної з версій та в якому присутні малі та великі зміщення. Даний датасет не використовувався для тренування, оскільки персонажі, не зовсім реалістичні.

FlyingChairs [32] датасет – перший датасет з достатньо великою вибіркою (більше 22 тисяч), в якому на певному фоні виставлялися стільці 62 типів. Недоліком даного датасету є однотипність об'єктів на зображенні, тому ми тренували на датасеті FlyingThings3D.

FlyingThings3D [15] – це синтетичний датасет з понад 22 тис. стерео пар для тренування та приблизно 4 тис. для тестування. В ньому на фоні коробок розташовані різні літаючі об'єкти (літаки, машини, дивани). Даний датасет має переваги: великий та має різні типи об'єктів. Тому спочатку ми тренували на ньому.

Пізніше ми звернули увагу на ще два датасети: Monkaa та Driving. Monkaa датасет має фон з різних пір року та незвичайні кольори. Також оскільки ми збиралися відправляти результати визначання глибини на Kitti веб-сторінку [36],

то для того, щоб мережа навчилася розпізнавати дорожні ситуації, ми взяли для тренування також датасет Driving. Даний датасет складався з 4 тисяч стерео зображень.

Всі три датасети (FlyingThings3D, Monkaa, Driving) ми об'єднали в один і назвали Синтетичний датасет, з кількістю стерео пар понад 36 тисяч. Даного обсягу достатньо для тренування глибокої згорткової мережі, щоб не було overfitting під час тренування.

3.4.2 Метрика

Для оцінки навчання нейронної мережі на вищенаведених датасетах, використовувалися наступні дві метрики:

а) Три пікселі помилки (3 PE);

Дана метрика була взята, оскільки вона використовується для оцінки результатів на Kitti [1] в категорії Stereo 2015 та Stereo 2012. Дана метрика використовувалася для оцінки отриманих результатів. (Також додатково до даної метрики використовувалася умова з Kitti вебсайту – відносна помилка). Піксель вважається помилково порахованим, якщо (формула 3.1):

$$\text{---} \quad (3.1)$$

б) MAE.

Для обчислення середнього використовують формулу 3.2.

—

(3.2)

Дана метрика використовувалася під час навчання нейронної мережі.

Висновки до розділу

Отже, з використанням згорткових нейронних мереж вдалося створити архітектуру, що б могла оцінювати невідповідність між двома картинками стереопари. Також в якості метрик, було обрано середнє значення модулів помилок (попільсьельно) та три пікселі помилки.

4 РЕЗУЛЬТАТИ ПОБУДОВИ МЕТОДУ ОБЧИСЛЕННЯ КАРТИ ГЛИБИНИ

4.1 Результати тренування

Архітектура нашої мережі була розроблена на фреймворку Caffe [37]. В якості оптимізатора використовувався Adam [38] з параметрами та .

Кількість зображень для тренування (batch size) був 4. Також під час тренування застосовувався метод multistep, з кроком в 70 тис. ітерацій. Початковий learning rate значення було встановлено в , зменшувався вдвічі до .

Для того щоб не було overfitting, під час тренування застосовувалась data augmentation, а саме поворот, обрізання, масштабування зображення, та зміна кольору (контрастність, гамма фактор зображення і освітленість). Таким чином вдалося значно збільшити розмір навчальних даних.

Тренування здійснювалося в 6 етапів. Кожний етап мав певні вагові коефіцієнти (рисунок 4.1), які задавалися для кожного проміжного та кінцевого шару втрат.

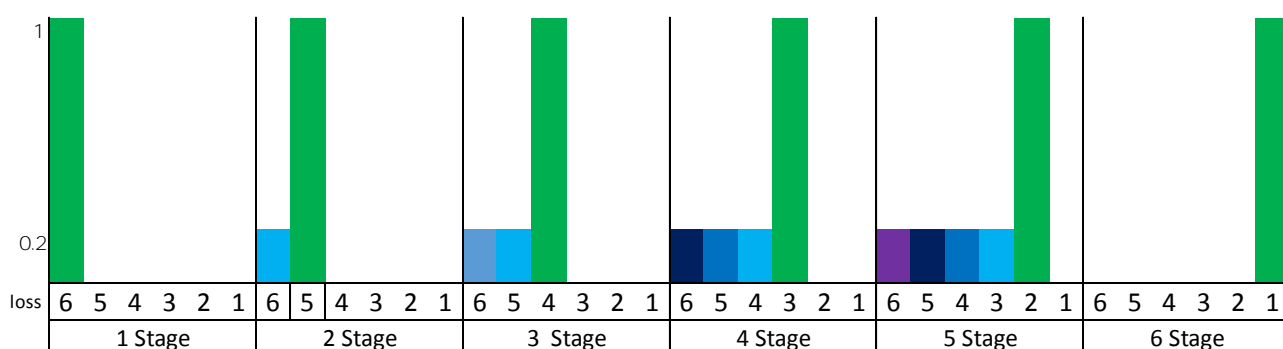


Рисунок 4.1 – Графік зміни коефіцієнтів шарів втрат

Тренування в 6 етапів дало змогу краще натренувати середні шари, ніж якби тренувати повністю кінець-в-кінець. На кожному етапі, якщо значення коефіцієнту для шару втрат був нуль, то даний шар не навчався. В останній стадії

навчання, для останнього (виходу) мережі було встановлено коефіцієнт 1, тоді як для інших був нуль, оскільки це останній етап, який займав найбільшу кількість часу і мав найпріоритетніше значення.

4.2 Порівняння з іншими методами

Порівняння DispNetEqual з іншими методами наведено в таблиці 2.

Таблиця 4.1 – Порівняння з іншими методами

| Датасет | Частина | | | | | MAE | 3PE |
|---------|---------|--|--|--|--|------------|------|
| | | | | | | | |
| | | | | | | - | - |
| | | | | | | / | / |
| | | | | | | - | - |
| | | | | | | / | 6.34 |
| Час(мс) | | | | | | 5 6 | |

Як бачимо з таблиці 4.1, наша мережа переважає архітектуру DispnetCorr1D: на Synthetic датасеті: на 1.83% з 3PE та на Kitti2015 датасеті: на 0.66% з 3PE.

Також наша архітектура має більший час рахування невідповідності, оскільки в кінці було додані шари. Час рахування невідповідності збільшився на 8 мс.

Також результат нашої архітектури були подані на KITTI змагання. Наша архітектура показала результат кращий на 0.55%, ніж оригінальна архітектура DispNetCorr1D. Наш результат на KITTI становить 3.79% D1-all помилка.

Порівняння DispnetCorr1D з DispnetEqual наведено в таблиці 4.2 .

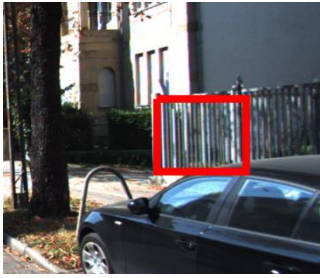
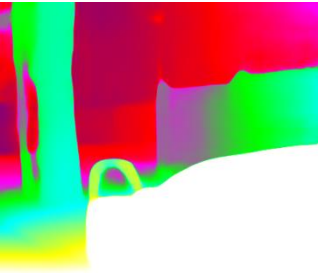
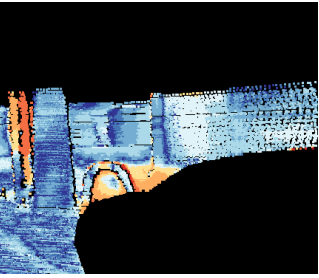
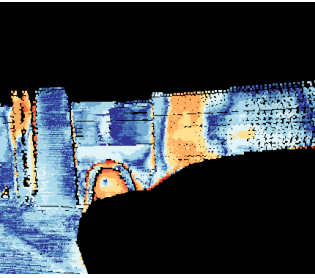
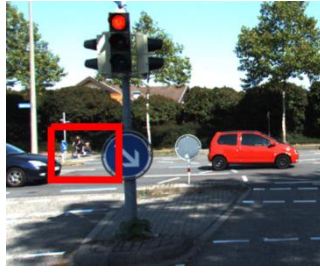
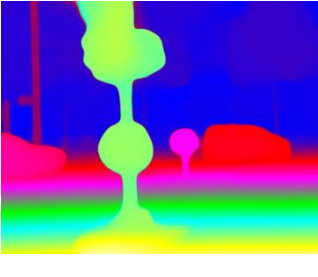
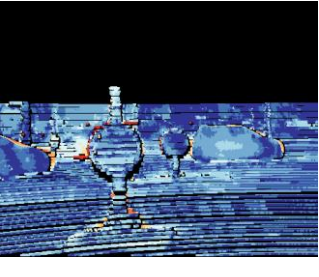
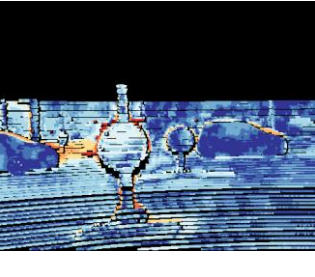

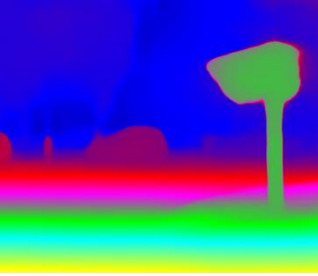
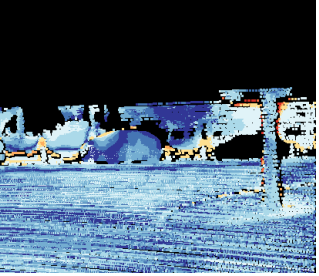
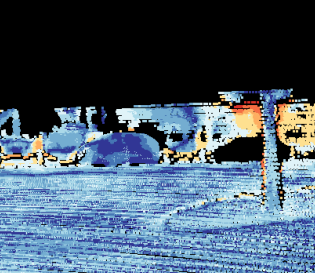
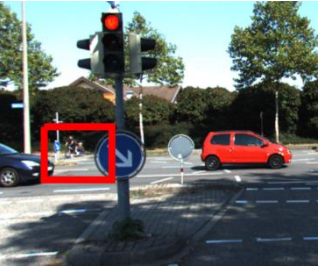
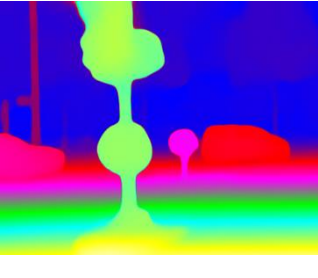
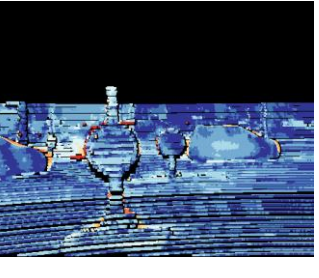
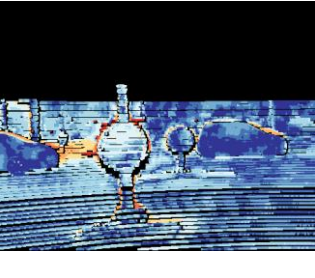

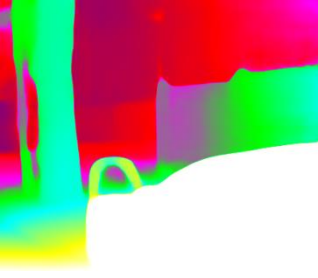
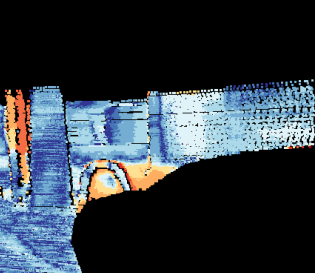
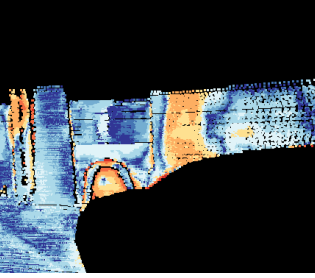
На зображеннях помилок: наявність червоного кольору означає помилку. Чим червоніший колір, тим помилка більша. Чим синіший колір, тим правильніше ми оцінили невідповідність.

Як бачимо, результат DispnetEqual має кращу невідповідність на задньому фоні зображення за DispnetCorr1D.

При експерименті з більшою кількістю шарів у кінці, коли зображення на початку збільшується вдвічі і в кінці зменшується за допомогою інтерполяції до вихідного розміру, було помічено, що час роботи мережі значно збільшувався, оскільки згорткові шари дуже чутливі до вхідного розміру, тому час рахування невідповідності збільшувався майже вдвічі. Через обмежений час використання GPU ресурсів, дану мережу не тренували.

При спробі додати ще один шар у глибину (всередину архітектури енодер-декодер) було помічено, що результат змінюється мало, оскільки на найнижчому рівні зображення буде зменшено аж в 128 раз, за кожним розміром. Оскільки ми спочатку тренували на Synthetic датасеті, де зображення розміру 540x960, то при такому зменшенні, на нижньому рівні буде дуже мала карта ознак.

Таблиця 4.2 – Графічне порівняння DispnetCorr1D з DispnetEqual

| | | | |
|---|---|--|---|
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Ліве зображення | DispnetEqual невідповідність | Помилка | Помилка |

Висновки до розділу

Використання згорткових нейронних мереж досягли визначних результатів в області комп'ютерного зору, а саме у визначенні глибини карти зі стерео пари. Але визначення глибини і досі потребує покращення. Ми запропонували використовувати згорткову нейронну мережу на типом архітектури енкодер-декодер, DispnetEqual. Дана мережа має додаткові шари, які збільшують невідповідність до вихідного розміру, на відміну від DispnetCorr1D. Наші експерименти показали, що використання додаткових шарів дозволяє значно покращити точність глибини, при цьому час отримання глибини становить 68 мс. Дана архітектура показала гарні результати на Kitti онлайн змаганні.

5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

Розробка та запуск стартап проекту на ринку передбачає низку кроків, що визначають перспективи ринку, графік та принципи організації виробництва, фінансовий аналіз та аналіз ризиків та заходи щодо сприяння пропозиції інвесторам.

Для будь-якого стартап проекту, найважливішим є успішний запуск на ринок. Спеціалісти у створенні та розробці проектів запуску окремо зазначають, що відсутність маркетингових знань та навичок, що дають можливість розробки ринкового проекту від початкової ідеї, є основною причиною високого рівня банкрутств компаній-початківців, і ця проблема може бути вирішено шляхом підготовки винахідників. Відповідно основною метою цього розділу є отримання знань про сутність, основні принципи розробки стратегії впровадження ринку та управління маркетингом інноваційних пускових проектів у галузях економіки, використання ефективних маркетингових інструментів для просування високотехнологічних товарів і послуг [41].

У цьому розділі ми проведемо перший етап розвитку стартап проекту, а саме - маркетинговий аналіз - ми відкриємо для себе ринкові можливості для використання результатів роботи розробленого науково-технічного рішення.

5.1 Опис ідеї проекту

Основна ідея проекту замінити використання лазерних систем з визначення глибини у засобів, що рухаються на дорозі на систему, що здійснює обчислення глибини на основі методу фотограмметрії. Зміст ідеї та напрямки застосування наведено в таблиці 5.1 .

Таблиця 5.1 – Опис ідеї стартап-проекту

| Зміст ідеї | Напрямки застосування | Вигоди для користувача |
|---|--|------------------------|
| Заміна на дешеві системи для визначення глибини у засобів що рухаються по дорозі замість лазерних систем. | Галузь автомобілебудування, що рухаються без водія. Системи асистенту авто. Військова сфера (транспортні засоби) | Економічні |

Система дозволить визначати глибину, дистанцію до об'єктів зовнішнього середовища (дорожньої обстановки).

Техніко-економічні характеристики ідеї проекту та їх оцінювання наведено в таблиці 5.2 .

Таблиця 5.2 – Сильні, слабкі та нейтральні характеристики ідеї проекту

| № п/ п | Техніко- економічні характеристи ки ідеї | (потенційні) товари/концепції конкурентів | | | W (слабк а сторон а) | N (нейт- ральна сторон а) | S (сильн а сторон а) |
|--------------|---|--|--------------------------|----------------------------|----------------------------------|---------------------------------------|----------------------------------|
| | | Мій проект | Конку- рент 1 | Конку- рент 2 | | | |
| 1. | Тренд споживання | Система визначення глибини за допомогою фотограммет рії | velody ne hdl- 64e | Velody ne VLP- 16 | | + | |
| 2. | Швидкість детектування | | | | | | + |
| 3. | Надійність | | | | | + | |
| 4. | Вартість підготовки та впровадженн я | | | | | | + |
| 5. | Вартість обслуговува ння | | | | | + | |
| 6. | Вартість продукту | | | | | | + |

5.2 Технологічний аудит ідеї проекту

Технологічна здійсненість ідеї проекту наведена в таблиці 5.3 .

Таблиця 5.3 – Технологічна здійсненність ідеї проекту

| <i>n/n</i> | <i>Ідея проекту</i> | <i>Технології її реалізації</i> | <i>Наявність технологій</i> | <i>Доступність технологій</i> |
|------------|------------------------------|---------------------------------|-----------------------------|-------------------------------|
| 1. | Визначення глибини | Deep Learning | + | + |
| 2. | Швидкість обчислення глибини | GPU | + | + |

5.3 Аналіз ринкових можливостей запуску стартап-проекту

У контексті конкретної інвестиційної ситуації вирішено завдання аналізу ринкових можливостей, а саме аналіз мікросередовища та макроекологічного середовища проекту, аналіз роздрібних та оптових ринків.

Під мікропідприємством проекту розуміється сукупність відносин в організації, яка реалізує інвестиційний проект між організацією та клієнтами, організацією та конкурентами. З точки зору інвестиційного проекту, аналіз власної діяльності має здійснюватися, якщо цей проект спрямований на розширення діяльності вже діючої організації через глибоке проникнення існуючого ринку, розширення ринкових меж з існуючими продуктом, випуск нового продукту на вже існуючому ринку або диверсифікацію його виробництва, а також аналіз діяльності конкурентів та потенційних покупців. Для аналізу власної діяльності використовуються наступні показники: дані про поточні

продажі, обсяги запасів на складах, канали руху товарів, регіони продажів, витрати, потік коштів, графік надходження товарів на склади [40].

Для оцінки діяльності на ринку фірм-конкурентів використовуються наступні показники: обсяг реалізованої продукції, частка в загальному обсязі продажів, характер продукції, що випускається (технічні характеристики, ціна, новизна, наявність послуг), практика рекламної діяльності; практика руху товарів (наявність складів, види транспорту, робота з дилерами та дистриб'юторами), маркетингова діяльність компанії (асортиментна політика, напрямки розвитку нових продуктів, маркетингова політика, методи інтенсифікації продажів, ціна політика та тенденції їх зміни), рівень виробничих витрат та шляхи їх зменшення, фінансове становище, кількісні показники діяльності (обсяг виробництва, інвестиції, витрати на науково-дослідну роботу, ступінь завантаження виробничих потужностей для виробництва конкуруючі продукти).

Для аналізу потенційних покупців використовуються наступні показники: ринкова ситуація, частка у загальному споживанні товарів, основні постачальники продукції, вимоги компанії до продукції, організаційна структура, торгова спроможність, роздрібні методи, умови надання пільг покупцям та постачальникам. , ефективність каналів збуту, загальний обсяг продажів, рентабельність торгових операцій, витрати на продаж, витрати на накопичення запасів, комісійні, отримані фірмою через посередництво.

Макроекономічне середовище інвестиційного проекту є факторами, які впливають на всі елементи мікросередовища проекту. Серед них - загальні чинники: демографічні показники (фертильність, середні сімейні показники, показники міграції, рівень освіти тощо), економічний (рівень економічного розвитку країни, заробітна плата, рівень безробіття, структура розподілу доходів між різними групами населення, структура витрати на споживання та ін)), природні (наявність і можливість раціонального використання сировини та енергоресурсів, необхідність охорони навколишнього середовища), науково-

технічний прогрес (напрями розвитку, організації, що проводять перспективні дослідження, можливість використання фундаментальних досліджень в прикладному плані тощо), політичні (законодавча база, ступінь державного контролю за дотриманням нормативних актів, ступінь впливу на політичну ситуацію громадських організацій та об'єднань), культурне середовище (тенденції в культурному середовищі та можливість націлювання їх організація в ході реалізації проекту).

Аналіз тенденцій та процесів у макроекономічному середовищі дозволяє визначити їх можливий вплив на реалізацію проекту. Завдання управління проектом - мінімізувати неконтрольовані фактори навколишнього середовища та знайти можливість впливати на контрольовані фактори.

Аналіз роздрібних та оптових ринків. Ринок роздрібних або споживчих товарів для ринку споживчих товарів та послуг для особистого споживання. Процес загального дослідження роздрібного ринку повинен включати визначення самого ринку (насамперед, товарів та покупок цих товарів, які відрізняються у доходах і споживанні, соціальному статусі, національності, культурних традиціях) та факторами, що визначають поведінка покупця при здійсненні покупки (економічні, науково-технічні, політичні, фактори культурного середовища). Процес, за яким покупець вирішує придбати товар, - це зрозуміти потребу продукту, пошук інформації про продукт, вибір доступної покупки, вибір покупки та відповідь на покупку.

Знання характеристик роздрібного ринку та фактори, що визначають поведінку покупця, дають можливість вплинути на прийняття покупцем рішення про покупку товару та визначення кількості майбутніх продажів проекту.

Характеристика клієнтів (потенційних) стартап-проекту наведено в таблиці 5.4

Таблиця 5.4 – Характеристика потенційних клієнтів стартап-проекту

| № п / п | Потреба, що формує ринок | Цільова аудиторія (цільові сегменти ринку) | Відмінності у поведінці різних потенційних цільових груп клієнтів | Вимоги споживачів до товару |
|------------------|--------------------------------------|---|---|---|
| 1 | Знання відстані до об'єктів | Виробники транспортн их засобів; ремонтні системи | Клієнти зможуть використовувати дану систему для визначення відстані. Також можуть удосконалювати продукт на свої тренувальних даних | До продукції: висока точність; зрозумілий інтерфейс; До компанії постачальника: Технічна підтримка, врахування побажань. |

Фактори загроз та їх зміст наведено в таблиці 5.5 .

Таблиця 5.5 – Фактори загроз

| п/п | Фактор | Зміст загрози | Можлива реакція компанії |
|-----|--|---|---|
| 1. | Система буде неточна в нових умовах | Неможливість використання через ненадійність. | Зняти вибірку в нових умовах; підправити продукт згідно заданих умов. |

Фактори можливостей наведено в таблиці 5.6 .

Таблиця 5.6 – Фактори можливостей

| п/п | Фактор | Зміст можливості | Можлива реакція компанії |
|-----|-------------------------------|--|--|
| 1 | Потреба в дешевому рішенні | Не має коштів на точні лазерні прилади | Збільшити ринок споживачів. |
| 2 | Новий гравець ринку | Нова компанія з продажу транспортних засобів | Пропонувати свій продукт та його можливості |

Ступеневий аналіз конкуренції ринку наведено в таблиці 5.7 .

Таблиця 5.7 – Ступеневий аналіз конкуренції на ринку

| Особливості конкурентного середовища | В чому проявляється дана характеристика | Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною) |
|--------------------------------------|---|--|
| Тип конкуренції - Чиста | На ринку поки немає подібних технологій | Необхідно стежити за виходами стартапів, швидко реалізовувати нові ідеї, а також слідкувати за іноземним ринком |
| Рівень конкурентної боротьби | Національний рівень конкурентної боротьби | Відповідність технології та якості продукту вимогам законодавства України |
| За галузевою ознакою | Міжгалузева | Урахування особливостей конкурентного середовища у кожній галузі, з якою взаємодіє стартап. |
| Конкуренція за видами товарів | Товарно-видова конкуренція | Відстеження тенденцій на ринку з можливістю появи на ринку продуктів-замінників. |
| Особливості конкурентного середовища | В чому проявляється дана характеристика | Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною) |
| За характером конкурентних переваг | Цінова | Удосконалення технології, що спрямована на підвищення базових переваг. |
| За інтенсивністю | Не марочна інтенсивність | Забезпечення масштабованості стартапу в найближчій перспективі для створення стійкого сприйняття стартапу як окремої бізнес одиниці. |

Аналіз конкуренції в галузі за Портером М. наведено в таблиці 5.8 .

Таблиця 5.8 – Аналіз конкуренції в галузі за М. Портером

| | Прямі конкуренти в галузі | Потенційні конкуренти | Постачальники | Клієнти | Товари-замінники |
|------------------|--|---|---|--|--|
| Складові аналізу | Навести перелік прямих конкурентів | Визначити бар'єри входження в ринок | Визначити фактори сили постачальників | Визначити фактори сили споживачів | Фактори загроз з боку замінників |
| Висновки: | Velodyne; LeddarTech; Innoviz; Trilumina; | Бар'єрів виходу на ринок не має, проте важливо переконати в потенційних клієнтів в тому що наш продукт працює | Заміна дорогого обладнання на більш дешеве. | Клієнти диктують умови своїми бажаннями щодо інтерфейсу програми | Наш товар і є замінником. Тому важливо щоб не вийшов продукт, схожий на наш. |

Обґрунтування факторів конкурентоспроможності наведено в таблиці 5.9 .

Таблиця 5.9 – Обґрунтування факторів конкурентоспроможності

| п/п | Фактор конкурентоспроможності | Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим) |
|-----|-------------------------------|---|
| 1 | Економічний | Низька вартість впровадження та експлуатації. |
| 2 | Надійність | Система у режимі онлайн виявляє непередбачувані ситуації. |
| 3 | Ергономічний | Підвищує зручність спостережень за об'єктами. |
| 4 | Багатофункціональність | Допоміжний засіб для іншого функціоналу (моделювання перешкод) |

Порівняльний аналіз сильних та слабких сторін у порівнянні з конкурентами наведено в таблиці 5.10 .

Таблиця 5.10 – Порівняльний аналіз сильних та слабких сторін

| № п/ п | Фактор конкурентоспроможності | Бали 1-20 | Рейтинг товарів-конкурентів у порівнянні з системою | | | | | | |
|--------------|-------------------------------|--------------|---|----|----|---|----|----|----|
| | | | -3 | -2 | -1 | 0 | +1 | +2 | +3 |
| 1 | Економічний | 18 | | | | | | | + |
| 2 | Надійність | 17 | | | | | + | | |
| 3 | Ергономічний | 18 | | | | | | | + |
| 4 | Багатофункціональність | 19 | | | | | | + | |

SWOT-аналіз стартап-проекту наведено в таблиці 5.11 .

Таблиця 5.11 – SWOT-аналіз стартап-проекту

| | |
|---|---|
| <p>Сильні сторони:</p> <p>Потреба в продукті;</p> <p>Зважена цінова політика;</p> <p>Врахування потреб споживачів.</p> | <p>Слабкі сторони:</p> <p>Частка ринку;</p> <p>Недостатні фінансові ресурси;</p> <p>Можлива ненадійність в нових умовах</p> |
| <p>Можливості:</p> <p>Зростання грошових доходів;</p> <p>Застосування сучасних технологій</p> <p>Швидке поширення на ринок</p> <p>Долучення нових гравців ринку</p> | <p>Загрози:</p> <p>Мала надійність;</p> <p>Нерозуміння споживачів вигоди;</p> |

Альтернативи ринкового впровадження стартап-проекту наведено в таблиці 5.12 .

Таблиця 5.12 – Альтернативи ринкового впровадження стартап-проекту

| /п | Альтернатива (орієнтовний комплекс заходів) ринкової поведінки | Ймовірність отримання ресурсів | Строки реалізації |
|----|---|--------------------------------|-------------------|
| . | Стратегія нейтралізації ринкових загроз сильними сторонами стартапу. | вище середньої | 1 рік |
| . | Стратегія підсилення сильних сторін за рахунок ринкових можливостей. | висока | 4 місяці |
| . | Стратегія компенсації слабких сторін наявними ринковими можливостями. | середня | 3 роки |

5.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів. Вибір цільових груп потенційних споживачів наведено в таблиці 5.13 .

Таблиця 5.13 – Вибір цільових груп потенційних споживачів

| № п/п | Опис профілю цільової групи потенційних клієнтів | Готовність споживачів сприйняти продукт | Орієнтовний попит в межах цільової групи | Інтенсивність конкуренції в сегменті | Простота входу у сегмент |
|----------|---|--|--|---|----------------------------------|
| 1. | Автомобільні компанії | висока | достатній | висока | високий бар'єр входу у галузь |
| 2. | Системи забезпечення обладнанням автомобілів | висока | достатній | помірна | високий бар'єр входу у галузь |

Визначення базової стратегії розвитку наведено в таблиці 5.14 .

Таблиця 5.14 – Визначення базової стратегії розвитку

| | | |
|---|---------------------------------|---|
| Обрана альтерна тива розвитку проекту | Стратегія охоплення ринку | Ключові конкурентоспроможні позиції відповідно до обраної альтернативи |
|---|---------------------------------|---|

Продовження таблиці 5.14

| | | |
|----------------------------------|--|---|
| Стратегія лідерства за витратами | Вибіркова та ексклюзивна стратегія охоплення ринку | <p>- цінова конкуренція між продавцями особливо сильна;</p> <p>- продукти, вироблені в галузі, мають високий ступінь стандартизації, бренди різних виробників практично ідентичні, і покупці не бачать значної різниці між ними;</p> <p>- якщо вартість споживачів переходить від одного продукту до іншого є відносно низькою, то це дозволяє їм вибирати товар за більш низькою ціною;</p> <p>- якщо зусилля значної частини покупців спрямовані на узгодження ціни, то товари з низькими цінами будуть реалізовуватися швидше.</p> <p>Конкурентоспроможне лідерство є серйозною конкурентною перевагою. Але водночас необхідно враховувати можливі ризики цієї стратегії.</p> <p>1. У випадку, коли конкуренти починають використовувати нову технологію, що відкриє свій шлях до зниження витрат, фірма-лідер може втратити свою перевагу, не отримуючи назад від попередніх інвестицій.</p> <p>2. Конкуренти можуть копіювати лідерські навички за рахунок коштів, що може суттєво скоротити термін прибутків.</p> <p>3. Працюючи над проблемою скорочення витрат, фірма може не помітити інших можливостей, які б дозволили йому стати лідерами.</p> <p>3. Працюючи над проблемою скорочення витрат, фірма може не помітити інших можливостей, які б дозволили йому стати лідерами.</p> <p>4. Зміна смаків та уподобань покупців також може позбавити фірму отриманих пільг</p> |
|----------------------------------|--|---|

Можливими шляхами досягнення переваг вартості є: спрощення розробки продукту; випуск продукту без зайвих покращень обладнання; реконструкція основних виробничих процесів; використання простішої і більш ефективної технології; відмова від використання дорогих матеріалів та комплектуючих; переміщення виробництва ближче до споживача / постачальника, щоб скоротити транспортні витрати тощо.

Лідерами з точки зору витрат є ті фірми, які постійно використовують всі способи заощаджувати гроші на всіх рівнях цілі. Низькі витрати дозволяють компанії не тільки встановлювати низькі ціни, отримуючи більші прибутки, ніж конкуренти, але також створюють перешкоди для захисту своїх позицій від дій п'яти конкурентних сил.

Визначення базової стратегії конкурентної поведінки наведено в таблиці 5.15 .

Таблиця 5.15 – Визначення базової стратегії конкурентної поведінки

| № п/п | Чи є проект «першо-прохідцем» на ринку? | Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів? | Чи буде компанія копіювати основні характеристики товару конкурента, і які? | Стратегія конкурентної поведінки* |
|-------|---|--|--|-----------------------------------|
| 1. | так | так | Так, основною задачею проекту являється збереження клієнтів. Принцип дії лишається не змінний. Змінюються алгоритми оптимізації | оборонна |

5.5 Розроблення маркетингової програми стартап-проекту

Маркетингова програма призначена для систематичного впровадження, об'єднаного єдиним цільовим та залежним від часу набором взаємозв'язаних завдань та вирішенням соціальних, економічних, науково-технічних, виробничих, організаційних, визначених ресурсів, а також джерел отримання цих ресурсів. .

Ефективний маркетинг починається з добре обізнаної, добре обізнаної маркетингової стратегії. Хороша стратегія допоможе вам визначити своє бачення, місію та бізнес-цілі, а також описати кроки, які вам необхідно вжити для досягнення цих цілей.

Маркетингова стратегія визначає загальний напрям і цілі вашого маркетингу, тому відрізняється від маркетингового плану, який окреслює конкретні дії, які необхідно вжити для реалізації маркетингової стратегії. Ваша маркетингова стратегія може бути розроблена протягом найближчих кількох років, тоді як ваш маркетинговий план зазвичай описує тактику, яку потрібно досягти цього року.

Визначення ключових переваг концепції потенційного товару наведено в таблиці 5.16 .

Таблиця 5.16 – Ключові переваги концепції потенційного товару

| № п/п | Потреба | Вигода, яку пропонує товар | Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити) |
|-------|-----------------|----------------------------|--|
| 1 | Зниження витрат | Економічна | Значно зменшує витрати, шляхом удосконалення алгоритму |

Концепція маркетингових комунікацій наведена в таблиці 5.17 .

Таблиця 5.17 – Концепція маркетингових комунікацій

| № п/ п | Специфіка поведінки цільових клієнтів | Канали комунікацій , якими користують ся цілові клієнти | Ключові позиції, обрані для позиціюван ня | Завдання рекламного повідомлення |
|--------------|--|--|---|--|
| 1 | Обережний вибір потенційних контрагентів, що зумовлено особливістю ринку | Інтернет- реклама | Технологія | Привернути увагу |

Висновки до розділу

На підставі проведених обчислень та вищевикладеного можна впевнено стверджувати, що в запропонований проект доцільно інвестувати кошти. З точки зору повернення коштів інвесторів проект є надзвичайно привабливим та ефективним за усіма характеристиками та показниками.

ВИСНОВКИ

У дисертаційній роботі одержано такі нові теоретичні та практичні результати:

а) удосконалено модель дорожньої обстановки, яка, на відміну від існуючих, має більш точно оцінює відстань до оточуючих об'єктів;

б) удосконалено метод отримання карти глибини, який, на відміну від існуючих, використовує більше даних з попередньо отриманих даних, наявністю кореляційного шару в архітектурі;

в) створено систему, яка оцінює глибину дорожньої обстановки на основі стерео пари з автомобіля.

Перспективою подальших досліджень є:

– використання більшої кількості навчальних даних, які можна отримати за допомогою симуляторів руху автомобіля;

– вдосконалити нейронну мережу за допомогою додаткового визначення контурів об'єктів;

– зменшити час роботи нейронної мережі за допомогою підходу «Вчитель-Студент».

ПЕРЕЛІК ПОСИЛАНЬ

1. Krizhevsky A. Imagenet classification with deep convolutional neural networks / A. Krizhevsky, I. Sutskever, G. Hinton. // Conference Advances in neural information processing systems. – 2012. – P. 1097–1105.
2. Deep Residual Learning for Image Recognition [Electronic resource] / K.He, X. Zhang, S. Ren, J. Sun // arXiv preprint arXiv:1512.03385. – 2015. – Mode of access: <https://arxiv.org/pdf/1512.03385v1.pdf>.
3. Scharstein D. High-accuracy stereo depth maps using structured light / D. Scharstein, R. Szeliski // In IEEE Computer Society Conference on Computer Vision and Pattern Recognition – 2003. – P. 195-202.
4. Scharstein D. Learning conditional random fields for stereo. / D. Scharstein and C. Pal. // In IEEE Conference on Computer Vision and Pattern Recognition – 2017 – P. 18 –25.
5. Hirschmüller H. Evaluation of cost functions for stereo matching. / H. Hirschmüller, D. Scharstein // In IEEE Conference on Computer Vision and Pattern Recognition – 2007. – P. 18-24.
6. High-resolution stereo datasets with subpixel-accurate ground truth / [Scharstein D., Hirschmüller H., Kitajima Y. et al.] // In German Conference on Pattern Recognition – 2014. – P. 31 – 42.
7. Sivaraman S. A review of recent developments in vision-based vehicle detection / S. Sivaraman and M. M. Trivedi. // IEEE Intelligent Vehicles Symposium – 2013. – P. 310 – 315.
8. Stereo vision based indoor/outdoor navigation for flying robots / Schmid K., Tomic T., Ruess F., Hirschmüller H. // IEEE/RSJ International Conference on Intelligent Robots and Systems. – 2013. – P. 3955 – 3962.

9. Zenati N. Dense stereo matching with application to augmented reality / N. Zenati, N. Zerhouni // In IEEE International Conference on Signal Processing and Communications – 2008. – P. 1503 –1506.
10. Shaked A. Improved stereo matching with constant highway networks and reflective confidence learning. / A. Shaked, L. Wolf // In IEEE Conference on Computer Vision and Pattern Recognition – 2017. – P. 6901 - 6910.
11. Zbontar J. Stereo matching by training a convolutional neural network to compare image patches/ J. Zbontar, Y. LeCun // Journal of Machine Learning Research – 2016. – vol. 17:2. – P.1-32.
12. End-to-end learning of geometry and context for deep stereo regression / [Kendall A., Martirosyan H., Dasgupta S. et al.]// In IEEE Conference on Computer Vision and Pattern Recognition – 2017. – P. 66–75.
13. FlowNet 2.0: Evolution of optical flow estimation with deep networks / [E. Ilg, N. Mayer, T. Saikia et al.] // In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition – 2017. – P. 2462–2470.
14. Scharstein D. High-accuracy stereo depth maps using structured light / D. Scharstein, R. Szeliski // In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2003). – 2013. – v. 1. – P.195-202.
15. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation / [N. Mayer, E. Ilg, P. Hausser et al] // In IEEE Conference on Computer Vision and Pattern Recognition – 2016. – P. 4040–4048.
16. H. Hirschmuller. Stereo processing by semiglobal matching and mutual information // IEEE Transactions on Pattern Analysis and Machine Intelligence – 2008. – P. 328–341.
17. Menze M. Object scene flow for autonomous vehicles. / M. Menze, A. Geiger. // In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition – 2015. – P. 3061 - 3070.

18. Правила дорожнього руху України [Електронний ресурс] /Верховна рада України // Постанова кабінету міністрів. – 2018. – Режим доступу до ресурсу: <http://zakon3.rada.gov.ua/laws/show/1306-2001-%D0%BF/paran16#n16>.
19. Hartley R. Multiple View Geometry in Computer Vision / R. Hartley, A. Zisserman. – New York: Cambridge University Press, 2003. – 655 p.
20. Kanadeet T. Development of a Video-Rate Stereo Machine / Kanadeet T., Kano H., Kimura S. // Proceedings of International Robotics and Systems Conference, 1995, v.3, P. 95 – 100.
21. Faugeras O. Real-time correlation-based stereo: algorithm, implementations and applications [Electronic resource] / O. Faugeras, T. Viéville, E. Theron // Inria. – 2013. – Mode of access: <https://hal.inria.fr/inria-00074658/file/RR-2013.pdf>.
22. Ansar A. Enhanced real - time stereo using bilateral filtering / A. Ansar, A. Castano, Matthies L.// Proceedings 2nd International Symposium on 3D Data Processing, Visualization and Transmission – 2004.
23. Hirschmüller H. Real-Time Correlation-Based Stereo Vision with Reduced Border Errors / Hirschmüller H, Innocent P., Garibaldi J // International Journal of Computer Vision – 2002. – No 1-3. – P. 229-246.
24. Yoon K. Locally adaptive support-weight approach for visual correspondence search / K. Yoon, I. Kweon // Computer Vision and Pattern Recognition – 2005 – P. 924-931.
25. Local stereo matching using Geodesic Support Weights // [Hosni A., Bleyer M., Gelautz M., Rhemann C.. // Conference ICIP – 2009.
26. Veksler O. Fast variable windows using integral images. / O. Veksler // Computer Vision and Pattern Recognition. – 2003. – P. 556-561.
27. Efficient Disparity Computation without Maximum Disparity for Real-Time Stereo Vision / C. Unger, S. Benhname, E. Wahl, N. Navab // British Machine Vision Conference. – 2009.

28. Sun J. Stereo matching using belief propagation / Sun J., Zheng N., Shum H. // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2003. – P. 787—800.
29. Matchnet: Unifying feature and metric learning for patch-based matching / [Han X., Leung T., Jia Y. et al.] // In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition – 2015. – P. 3279–3286.
30. Park H. Look wider to match image patches with convolutional neural networks / H. Park and K. M. Lee. // IEEE Signal Processing Letters, 2017, v. 24 – P. 1788 – 1792.
31. Luo W. Efficient deep learning for stereo matching / Luo W., Schwing A., Urtasun R. // IEEE Conference on Computer Vision and Pattern Recognition. – 2016. – P. 5695–5703.
32. Flownet: Learning optical flow with convolutional networks. / [A. Dosovitskiy, P. Fischer, E. Ilg et al.] // In Proc. of the IEEE International Conference on Computer Vision – 2015. – P. 2758–2766.
33. Godard C. Unsupervised monocular depth estimation with left-right consistency / C. Godard, O. Mac Aodha, G. J. Brostow // In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition – 2016. – P. 279–290.
34. Kuznetsov Y. Semi-supervised deep learning for monocular depth map prediction / Kuznetsov Y., Stücker J., Leibe B. // In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition – 2017. – P. 6647–6655.
35. A naturalistic open source movie for optical flow evaluation / D. J. Butler, J. Wulff, G. B. Stanley, M. J. Black. // Proceeding ECCV'12 Proceedings of the 12th European conference on Computer Vision, 2012, v. 6 – P. 611 – 625.
36. Geiger A. Are we ready for autonomous driving? / A. Geiger // IEEE Conference on Computer Vision and Pattern Recognition. – 2012. – P. 3354–3361.
37. Caffe: Convolutional architecture for fast feature embedding [Electronic resource] / [Y. Jia, E. Shelhamer, J. Donahue et al.] // ArXiv. – 2014. – Mode of access: <https://arxiv.org/abs/1408.5093>.

38. Kingma D. Adam: A method for stochastic optimization [Electronic resource] / D. Kingma, J. Ba // ArXiv. – 2014. – Mode of access: <https://arxiv.org/abs/1412.6980>.
39. Бернштейн Л. А. Анализ финансовой отчетности / Л. А. Бернштейн. // Москва: Финансы и статистика, 1996. – 622 с.
40. Герасимчук В.Г. Маркетинг: теорія і практика / Герасимчук В.Г. // Навч. посібник. - К.: Вища школа, 1994. - 327 с.
41. Zhong Y. Self-supervised learning for stereo matching with self-improving ability [Electronic resource] / Y. Zhong, Y. Dai, H. Li // Arxiv. – 2017. – Mode of access: <https://arxiv.org/abs/1709.00930>.

Додаток А Лістинг програми

Лістинг файлу model.prototxt

```
layer {
  name: "Image1"
  type: "ImageData"
  top: "img0"
  top: "label0"
  image_data_param {
    source: "tmp/img1.txt"
    batch_size: 1
  }
}

layer {
  type: "Silence"
  bottom: "label0"
}

layer {
  name: "Image2"
  type: "ImageData"
  top: "img1"
  top: "label1"
  image_data_param {
    source: "tmp/img2.txt"
    batch_size: 1
  }
}

layer {
  type: "Silence"
  bottom: "label1"
}

layer {
  name: "Eltwise2"
  type: "Eltwise"
  bottom: "img1"
  top: "img1s"
  eltwise_param {
    operation: SUM
    coeff: 0.00392156862745098
  }
}

layer {
  name: "Eltwise1"
  type: "Eltwise"
  bottom: "img0"
  top: "img0s"
  eltwise_param {
    operation: SUM
    coeff: 0.00392156862745098
  }
}
```

```

layer {
  name: "img0s_aug"
  type: "DataAugmentation"
  bottom: "img0s"
  top: "img0_nomean"
  augmentation_param {
    augment_during_test: true
    recompute_mean: 1000
    mean_per_pixel: false
    crop_width: $TARGET_WIDTH
    crop_height: $TARGET_HEIGHT
  }
}

```

```

layer {
  name: "img1s_aug"
  type: "DataAugmentation"
  bottom: "img1s"
  top: "img1_nomean"
  augmentation_param {
    augment_during_test: true
    recompute_mean: 1000
    mean_per_pixel: false
    crop_width: $TARGET_WIDTH
    crop_height: $TARGET_HEIGHT
  }
}

```

```

layer {
  name: "Resample1"
  type: "Resample"
  bottom: "img0_nomean"
  top: "img0_nomean_resize"
  resample_param {
    width: $ADAPTED_WIDTH
    height: $ADAPTED_HEIGHT
    type: LINEAR
    antialias: true
  }
}

```

```

layer {
  name: "Resample2"
  type: "Resample"
  bottom: "img1_nomean"
  top: "img1_nomean_resize"
  resample_param {
    width: $ADAPTED_WIDTH
    height: $ADAPTED_HEIGHT
    type: LINEAR
    antialias: true
  }
}

```

```

layer {
  name: "conv1"
  type: "Convolution"
  bottom: "img0_nomean_resize"
  bottom: "img1_nomean_resize"
  top: "conv1a"
  top: "conv1b"
  param {
    lr_mult: 1
  }
}

```



```

    decay_mult: 1
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 64
    pad: 3
    kernel_size: 7
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "ReLU2"
  type: "ReLU"
  bottom: "conv1b"
  top: "conv1b"
  relu_param {
    negative_slope: 0.1
  }
}

```

```

layer {
  name: "ReLU1"
  type: "ReLU"
  bottom: "conv1a"
  top: "conv1a"
  relu_param {
    negative_slope: 0.1
  }
}

```

```

layer {
  name: "conv2"
  type: "Convolution"
  bottom: "conv1a"
  bottom: "conv1b"
  top: "conv2a"
  top: "conv2b"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 128
    pad: 2
    kernel_size: 5
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {

```

```

        type: "constant"
    }
    engine: CUDNN
}
}

```

```

layer {
  name: "ReLU3"
  type: "ReLU"
  bottom: "conv2a"
  top: "conv2a"
  relu_param {
    negative_slope: 0.1
  }
}

```

```

layer {
  name: "ReLU4"
  type: "ReLU"
  bottom: "conv2b"
  top: "conv2b"
  relu_param {
    negative_slope: 0.1
  }
}

```

```

layer {
  name: "conv_redir"
  type: "Convolution"
  bottom: "conv2a"
  top: "conv_redir"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 64
    pad: 0
    kernel_size: 1
    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "corr"
  type: "Correlation1D"
  bottom: "conv2a"
  bottom: "conv2b"
  top: "corr"
  correlation_param {
    pad: 40
    kernel_size: 1
    max_displacement: 40
  }
}

```

```

    stride_1: 1
    stride_2: 1
  }
}

```

```

layer {
  name: "ReLU5"
  type: "ReLU"
  bottom: "conv_redir"
  top: "conv_redir"
  relu_param {
    negative_slope: 0.1
  }
}

```

```

layer {
  name: "Concat1"
  type: "Concat"
  bottom: "corr"
  bottom: "conv_redir"
  top: "blob16"
  concat_param {
    axis: 1
  }
}

```

```

layer {
  name: "conv3"
  type: "Convolution"
  bottom: "blob16"
  top: "conv3"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 256
    pad: 2
    kernel_size: 5
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "ReLU6"
  type: "ReLU"
  bottom: "conv3"
  top: "conv3"
  relu_param {
    negative_slope: 0.1
  }
}

```

```

layer {
  name: "conv3_1"
  type: "Convolution"
  bottom: "conv3"
  top: "conv3_1"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 256
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "ReLU7"
  type: "ReLU"
  bottom: "conv3_1"
  top: "conv3_1"
  relu_param {
    negative_slope: 0.1
  }
}

```

```

layer {
  name: "conv4"
  type: "Convolution"
  bottom: "conv3_1"
  top: "conv4"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 512
    pad: 1
    kernel_size: 3
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "ReLU8"
  type: "ReLU"
  bottom: "conv4"
  top: "conv4"
  relu_param {
    negative_slope: 0.1
  }
}

```

```

layer {
  name: "conv4_1"
  type: "Convolution"
  bottom: "conv4"
  top: "conv4_1"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 512
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "ReLU9"
  type: "ReLU"
  bottom: "conv4_1"
  top: "conv4_1"
  relu_param {
    negative_slope: 0.1
  }
}

```

```

layer {
  name: "conv5"
  type: "Convolution"
  bottom: "conv4_1"
  top: "conv5"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 512
  }
}

```

```

    pad: 1
    kernel_size: 3
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "ReLU10"
  type: "ReLU"
  bottom: "conv5"
  top: "conv5"
  relu_param {
    negative_slope: 0.1
  }
}

```

```

layer {
  name: "conv5_1"
  type: "Convolution"
  bottom: "conv5"
  top: "conv5_1"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 512
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "ReLU11"
  type: "ReLU"
  bottom: "conv5_1"
  top: "conv5_1"
  relu_param {
    negative_slope: 0.1
  }
}

```

```

layer {
  name: "conv6"
  type: "Convolution"

```

```

bottom: "conv5_1"
top: "conv6"
param {
  lr_mult: 1
  decay_mult: 1
}
param {
  lr_mult: 1
  decay_mult: 0
}
convolution_param {
  num_output: 1024
  pad: 1
  kernel_size: 3
  stride: 2
  weight_filler {
    type: "msra"
  }
  bias_filler {
    type: "constant"
  }
  engine: CUDNN
}
}

```

```

layer {
  name: "ReLU12"
  type: "ReLU"
  bottom: "conv6"
  top: "conv6"
  relu_param {
    negative_slope: 0.1
  }
}

```

```

layer {
  name: "conv6_1"
  type: "Convolution"
  bottom: "conv6"
  top: "conv6_1"
  param {
    lr_mult: 1
    decay_mult: 1
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 1024
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "ReLU13"

```

```

type: "ReLU"
bottom: "conv6_1"
top: "conv6_1"
relu_param {
  negative_slope: 0.1
}
}

```

```

layer {
  name: "deconv5"
  type: "Deconvolution"
  bottom: "conv6_1"
  top: "deconv5"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 0
    decay_mult: 0
  }
  convolution_param {
    num_output: 512
    pad: 1
    kernel_size: 4
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "Convolution1"
  type: "Convolution"
  bottom: "conv6_1"
  top: "predict_flow6"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 1
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {

```



```

name: "ReLU14"
type: "ReLU"
bottom: "deconv5"
top: "deconv5"
relu_param {
  negative_slope: 0.1
}
}

```

```

layer {
  name: "upsample_flow6to5"
  type: "Deconvolution"
  bottom: "predict_flow6"
  top: "upsampled_flow6_to_5"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 0
    decay_mult: 0
  }
  convolution_param {
    num_output: 1
    pad: 1
    kernel_size: 4
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "Concat2"
  type: "Concat"
  bottom: "conv5_1"
  bottom: "deconv5"
  bottom: "upsampled_flow6_to_5"
  top: "blob30"
}

```

```

layer {
  name: "Convolution2"
  type: "Convolution"
  bottom: "blob30"
  top: "concat5"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 512
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {

```

```

        type: "msra"
    }
    bias_filler {
        type: "constant"
    }
    engine: CUDNN
}
}

```

```

layer {
    name: "deconv4"
    type: "Deconvolution"
    bottom: "concat5"
    top: "deconv4"
    param {
        lr_mult: 1
        decay_mult: 0
    }
    param {
        lr_mult: 0
        decay_mult: 0
    }
    convolution_param {
        num_output: 256
        pad: 1
        kernel_size: 4
        stride: 2
        weight_filler {
            type: "msra"
        }
        bias_filler {
            type: "constant"
        }
        engine: CUDNN
    }
}

```

```

layer {
    name: "Convolution3"
    type: "Convolution"
    bottom: "concat5"
    top: "predict_flow5"
    param {
        lr_mult: 1
        decay_mult: 0
    }
    param {
        lr_mult: 1
        decay_mult: 0
    }
    convolution_param {
        num_output: 1
        pad: 1
        kernel_size: 3
        stride: 1
        weight_filler {
            type: "msra"
        }
        bias_filler {
            type: "constant"
        }
        engine: CUDNN
    }
}

```

```

layer {
  name: "upsample_flow5to4"
  type: "Deconvolution"
  bottom: "predict_flow5"
  top: "upsampled_flow5_to_4"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 0
    decay_mult: 0
  }
  convolution_param {
    num_output: 1
    pad: 1
    kernel_size: 4
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "ReLU15"
  type: "ReLU"
  bottom: "deconv4"
  top: "deconv4"
  relu_param {
    negative_slope: 0.1
  }
}

```

```

layer {
  name: "Concat3"
  type: "Concat"
  bottom: "conv4_1"
  bottom: "deconv4"
  bottom: "upsampled_flow5_to_4"
  top: "blob37"
}

```

```

layer {
  name: "Convolution4"
  type: "Convolution"
  bottom: "blob37"
  top: "concat4"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 256
    pad: 1
    kernel_size: 3
    stride: 1
  }
}

```

```

weight_filler {
  type: "msra"
}
bias_filler {
  type: "constant"
}
engine: CUDNN
}
}

```

```

layer {
  name: "deconv3"
  type: "Deconvolution"
  bottom: "concat4"
  top: "deconv3"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 0
    decay_mult: 0
  }
  convolution_param {
    num_output: 128
    pad: 1
    kernel_size: 4
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "Convolution5"
  type: "Convolution"
  bottom: "concat4"
  top: "predict_flow4"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 1
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "upsample_flow4to3"
  type: "Deconvolution"
  bottom: "predict_flow4"
  top: "upsampled_flow4_to_3"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 0
    decay_mult: 0
  }
  convolution_param {
    num_output: 1
    pad: 1
    kernel_size: 4
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "ReLU16"
  type: "ReLU"
  bottom: "deconv3"
  top: "deconv3"
  relu_param {
    negative_slope: 0.1
  }
}

```

```

layer {
  name: "Concat4"
  type: "Concat"
  bottom: "conv3_1"
  bottom: "deconv3"
  bottom: "upsampled_flow4_to_3"
  top: "blob44"
}

```

```

layer {
  name: "Convolution6"
  type: "Convolution"
  bottom: "blob44"
  top: "concat3"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 128
    pad: 1
    kernel_size: 3

```

```

    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "deconv2"
  type: "Deconvolution"
  bottom: "concat3"
  top: "deconv2"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 0
    decay_mult: 0
  }
  convolution_param {
    num_output: 64
    pad: 1
    kernel_size: 4
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "Convolution7"
  type: "Convolution"
  bottom: "concat3"
  top: "predict_flow3"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 1
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "upsample_flow3to2"
  type: "Deconvolution"
  bottom: "predict_flow3"
  top: "upsampled_flow3_to_2"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 0
    decay_mult: 0
  }
  convolution_param {
    num_output: 1
    pad: 1
    kernel_size: 4
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "ReLU17"
  type: "ReLU"
  bottom: "deconv2"
  top: "deconv2"
  relu_param {
    negative_slope: 0.1
  }
}

```

```

layer {
  name: "Concat5"
  type: "Concat"
  bottom: "conv2a"
  bottom: "deconv2"
  bottom: "upsampled_flow3_to_2"
  top: "blob51"
}

```

```

layer {
  name: "Convolution8"
  type: "Convolution"
  bottom: "blob51"
  top: "concat2"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 64
    pad: 1

```

```

    kernel_size: 3
    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "deconv1"
  type: "Deconvolution"
  bottom: "concat2"
  top: "deconv1"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 0
    decay_mult: 0
  }
  convolution_param {
    num_output: 32
    pad: 1
    kernel_size: 4
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```

```

layer {
  name: "Convolution9"
  type: "Convolution"
  bottom: "concat2"
  top: "predict_flow2"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 1
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

```



```
}
```

```
layer {
  name: "upsample_flow2to1"
  type: "Deconvolution"
  bottom: "predict_flow2"
  top: "upsampled_flow2_to_1"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 0
    decay_mult: 0
  }
  convolution_param {
    num_output: 1
    pad: 1
    kernel_size: 4
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}
```

```
layer {
  name: "ReLU18"
  type: "ReLU"
  bottom: "deconv1"
  top: "deconv1"
  relu_param {
    negative_slope: 0.1
  }
}
```

```
layer {
  name: "Concat6"
  type: "Concat"
  bottom: "conv1a"
  bottom: "deconv1"
  bottom: "upsampled_flow2_to_1"
  top: "blob58"
}
```

```
layer {
  name: "Convolution10"
  type: "Convolution"
  bottom: "blob58"
  top: "concat1"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 32
```

```

    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

layer {
  name: "Convolution11"
  type: "Convolution"
  bottom: "concat1"
  top: "predict_flow1"
  param {
    lr_mult: 1
    decay_mult: 0
  }
  param {
    lr_mult: 1
    decay_mult: 0
  }
  convolution_param {
    num_output: 1
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}

layer {
  name: "deconv0"
  type: "Deconvolution"
  bottom: "concat0"
  top: "deconv0"
  param {
    lr_mult: 0
    decay_mult: 0
  }
  param {
    lr_mult: 0
    decay_mult: 0
  }
  convolution_param {
    num_output: 16
    pad: 1
    kernel_size: 4
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
}
propagate_down: false

```

```

}
layer {
  name: "ReLU29"
  type: "ReLU"
  bottom: "deconv0"
  top: "deconv0"
  relu_param {
    negative_slope: 0.1
  }
}
layer {
  name: "upsample_flow1to0"
  type: "Deconvolution"
  bottom: "predict_flow1"
  top: "upsampled_flow1_to_0"
  param {
    lr_mult: 0
    decay_mult: 0
  }
  param {
    lr_mult: 0
    decay_mult: 0
  }
  convolution_param {
    num_output: 1
    pad: 1
    kernel_size: 4
    stride: 2
    weight_filler {
      type: "msra"
    }
    bias_filler {
      type: "constant"
    }
    engine: CUDNN
  }
  propagate_down: false
}
layer {
  name: "Concat8"
  type: "Concat"
  bottom: "img0_aug"
  bottom: "deconv0"
  bottom: "upsampled_flow1_to_0"
  top: "blob72"
}
layer {
  name: "Convolution12"
  type: "Convolution"
  bottom: "blob72"
  top: "concat0"
  param {
    lr_mult: 0
    decay_mult: 0
  }
  param {
    lr_mult: 0
    decay_mult: 0
  }
  convolution_param {
    num_output: 32
    pad: 1
    kernel_size: 3
    stride: 1
    weight_filler {
      type: "msra"
    }
    bias_filler {

```

```

        type: "constant"
    }
    engine: CUDNN
}
propagate_down: false
}
layer {
    name: "Convolution13"
    type: "Convolution"
    bottom: "concat0"
    top: "predict_flow0"
    param {
        lr_mult: 0
        decay_mult: 0
    }
    param {
        lr_mult: 0
        decay_mult: 0
    }
    convolution_param {
        num_output: 1
        pad: 1
        kernel_size: 3
        stride: 1
        weight_filler {
            type: "msra"
        }
        bias_filler {
            type: "constant"
        }
        engine: CUDNN
    }
    propagate_down: false
}
layer {
    name: "NegReLU7"
    type: "NegReLU"
    bottom: "predict_flow0"
    top: "blob76"
    relu_param {
        negative_slope: 0
    }
}

layer {
    name: "Resample4"
    type: "Resample"
    bottom: "predict_flow0"
    top: "predict_disp_resize"
    resample_param {
        width: $TARGET_WIDTH
        height: $TARGET_HEIGHT
        type: LINEAR
        antialias: true
    }
}

layer {
    name: "Eltwise4"
    type: "Eltwise"
    bottom: "predict_disp_resize"
    top: "predict_disp_final"
    eltwise_param {
        operation: SUM
        coeff: $SCALE_WIDTH
    }
}

```

```
layer {  
  name: "PFMWriter1"  
  type: "PFMWriter"  
  bottom: "predict_disp_final"  
  writer_param {  
    folder: "./output"  
    prefix: "disp-"  
    suffix: ""  
    scale: 1.0  
  }  
}
```

Додаток Б Ілюстративний матеріал

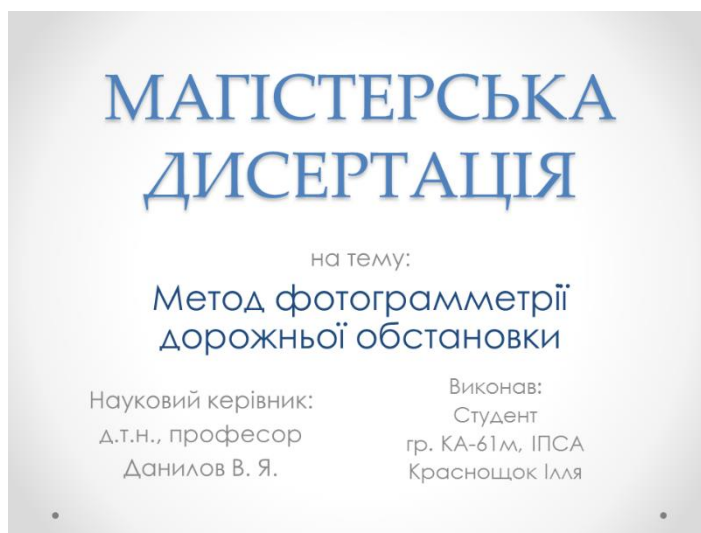


Рисунок Б.1 – Слайд 1

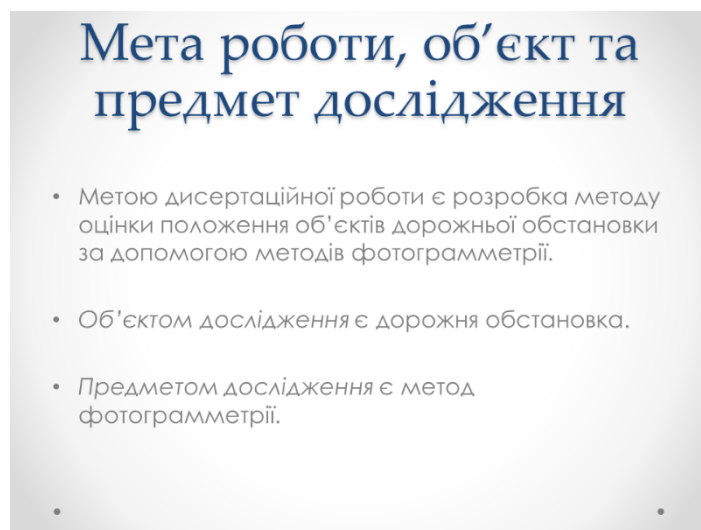


Рисунок Б.2 – Слайд 2

Задачі дослідження

- Провести аналіз методу фотограмметрії;
- Провести аналіз основних проблем обробки фотографій;
- Розробити метод для оцінки глибини дорожньої обстановки;
- Розробити систему для побудови карт глибини дорожньої обстановки зі стерео пар.

Рисунок Б.3 – Слайд 3

Метод фотограмметрії

- Наука, що дозволяє за допомогою фотографування, способів обробки знімків і спеціальних технологій отримувати зображення і визначати по ним просторове положення фізичних об'єктів на місцевості і їх характеристики має назву **фотограмметрія**.
- Окремим випадком фотограмметрії, є **стереофотограмметрія**.

Рисунок Б.4 – Слайд 4

Стандартні методи

- Алгоритм:

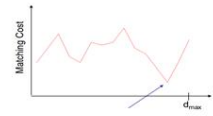
1) Попередня обробка

(Вирівнювання гістограм, Білатеральний фільтр)

2) Функція вартості

$$|I_L(x, y) - I_R(x + d, y)|;$$

$$(I_L(x, y) - I_R(x + d, y))^2$$



3) Сукупна функція вартості

$$\sum_{(x,y) \in W} |I_R(x, y) - I_L(x + d, y)|$$

$$\sum_{(x,y) \in W} (I_R(x, y) - I_L(x + d, y))^2$$



4) Підрахунок невідповідності

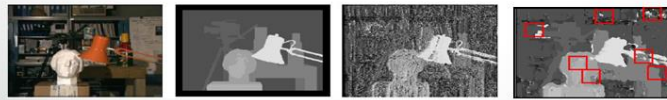


Рисунок Б.7 – Слайд 7

Модель

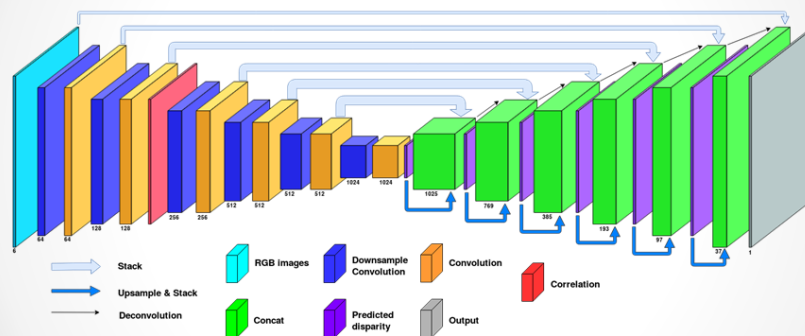


Рисунок Б.8 – Слайд 8

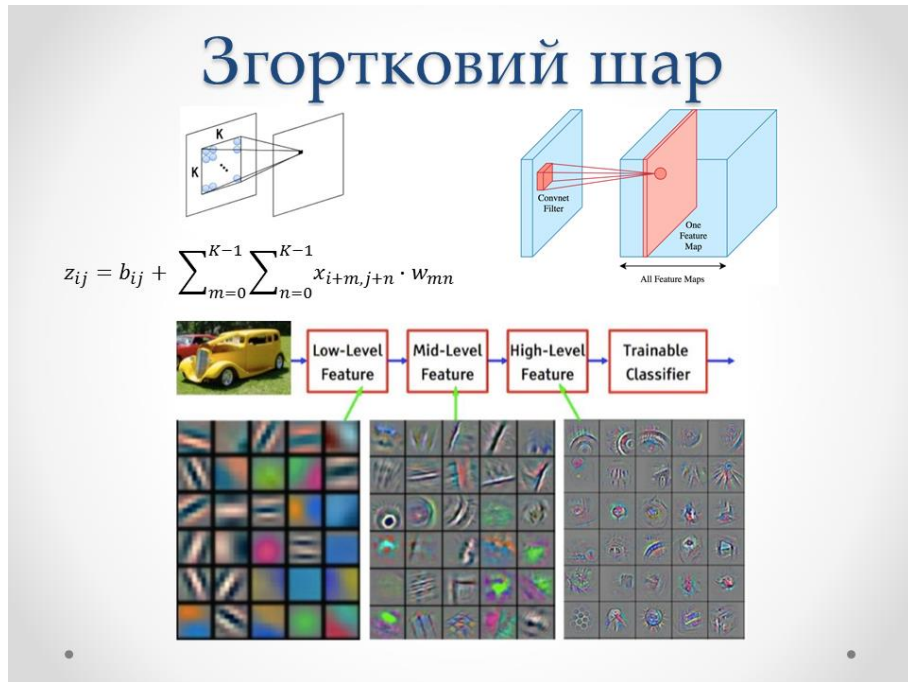


Рисунок Б.9 – Слайд 9

Кореляційний шар

- Нехай задано наступні дві карти ознак:

$$f_1, f_2: \mathbb{R}^2 \rightarrow \mathbb{R}^c,$$
 де w, h, c – це відповідно ширина, висота та кількість каналів.
- Даний кореляційний шар дозволяє порівнювати кожний патч з f_1 з кожним патчем з f_2 .
- Формула:

$$c(x_1, x_2) = \sum_{o \in [-k, k] \times [-k, k]} \langle f_1(x_1 + o), f_2(x_2 + o) \rangle$$

Рисунок Б.10 – Слайд 10

Датасети

- FlyingThings3D



- Monkaa



- Driving



:об'єднали в один (35 тис.)

- Kitti Stereo 2015
(200)

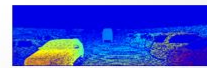


Рисунок Б.11 – Слайд 11

Метрики

- Три пікселі помилки (3 PE):

$$(|d_{gt} - d_{est}| > 3) \wedge \left(\frac{|d_{gt} - d_{est}|}{d_{gt}} > 0.05 \right)$$

- MAE:

$$MAE = \frac{1}{n} \sum_{i=1}^n |d_{gt} - d_{est}|$$

Рисунок Б.12 – Слайд 12

Тренування ЗНМ

- Метод оптимізації Adam

- Вагові коефіцієнти:

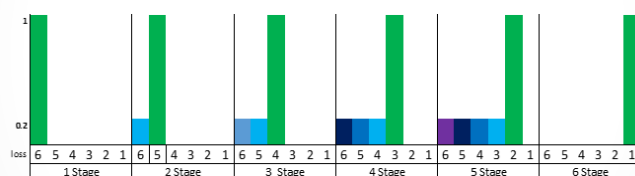


Рисунок Б.13 – Слайд 13

Результати

- Результат тренування на синтетичному датасеті:

| MAE | 3PE |
|------|------|
| 1.56 | 9.25 |

- Результат дотренування на Kitti датасеті:

| MAE | 3PE |
|------|------|
| 0.57 | 3.75 |

- Результат на KITTI становить **3.79%** D1-all помилка

| Error | D1-bg | D1-fg | D1-all |
|-----------|-------|-------|--------|
| All / All | 3.64 | 4.22 | 3.79 |

Рисунок Б.14 – Слайд 14

Порівняння з іншими методами

| Датасет | Частина | DispNetCorr1D | | DispNetEqual | | DeepCostAggr | |
|------------|---------|---------------|-------|--------------|------|--------------|------|
| | | MAE | 3PE | MAE | 3PE | MAE | 3PE |
| Synthetic | train | 1.48 | 9.76 | 1.23 | 8.28 | - | - |
| | test | 1.73 | 11.08 | 1.56 | 9.25 | - | - |
| Kitti 2015 | train | 0.47 | 3.41 | 0.34 | 3.19 | - | - |
| | test | 0.65 | 4.41 | 0.57 | 3.75 | - | 6.34 |
| Час(мс) | - | 60 | | 68 | | 34 | |

Рисунок Б.15 – Слайд 15

Порівняння з іншими методами

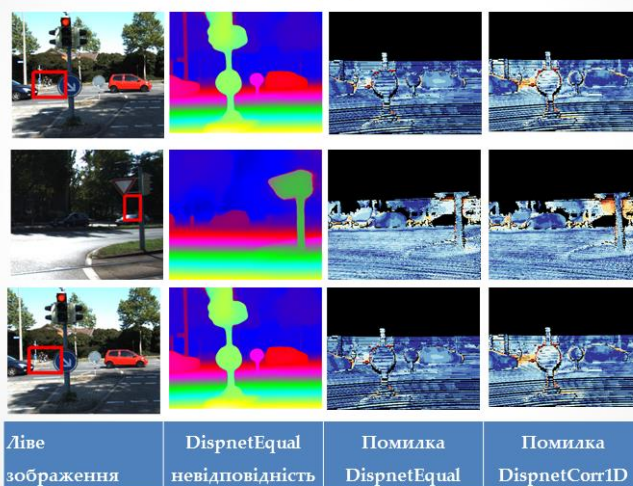


Рисунок Б.16 – Слайд 16

Система визначення глибини

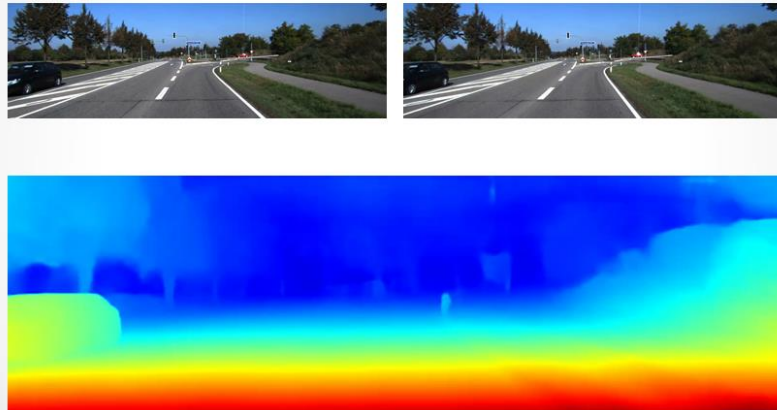


Рисунок Б.17 – Слайд 17

Висновки

- удосконалено модель дорожньої обстановки, яка, на відміну від існуючих більш точно оцінює відстань до оточуючих об'єктів;
- удосконалено метод отримання карти глибини, який, на відміну від існуючих, використовує більше даних з попередньо отриманих та наявності кореляційного шару в архітектурі;
- створено систему, яка оцінює глибину дорожньої обстановки на основі стерео пари з автомобіля.

Рисунок Б.18 – Слайд 18

Стаття

- За матеріалами магістерської дисертації опубліковано статтю в міжнародному журналі: «Енкодер-декодер згорткова нейронна мережа з кореляційним шаром для оцінки глибини»

Рисунок Б.19 – Слайд 19

Перспективи подальших досліджень

- використання більшої кількості навчальних даних, які можна отримати за допомогою симуляторів руху автомобіля;
- вдосконалити нейронну мережу за допомогою додаткового визначення контурів об'єктів;
- зменшити час роботи нейронної мережі за допомогою підходу «Вчитель-Студент».

Рисунок Б.20 – Слайд 20



Рисунок Б.21 – Слайд 21